Sun Java™ System

Sun Java Enterprise System 5 Upgrade Guide for UNIX

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

Part No: 819-6553-11
June 2007

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

THIS PRODUCT CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF SUN MICROSYSTEMS, INC. USE, DISCLOSURE OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF SUN MICROSYSTEMS, INC.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

This distribution may include materials developed by third parties.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, the Duke logo, the Java Coffee Cup logo, the Solaris logo, the SunTone Certified logo and the Sun ONE logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc.

Legato and the Legato logo are registered trademarks, and Legato NetWorker, are trademarks or registered trademarks of Legato Systems, Inc. The Netscape Communications Corp logo is a trademark or registered trademark of Netscape Communications Corporation.

The OPEN LOOK and Sun(TM) Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Products covered by and information contained in this service manual are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

DOCUMENTATION IS PROVIDED “AS IS” AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. détient les droits de propriété intellectuels relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés à l'adresse http://www.sun.com/patents et un ou des brevets supplémentaires ou des applications de brevet en attente aux Etats - Unis et dans les autres pays.

CE PRODUIT CONTIENT DES INFORMATIONS CONFIDENTIELLES ET DES SECRETS COMMERCIAUX DE SUN MICROSYSTEMS, INC. SON UTILISATION, SA DIVULGATION ET SA REPRODUCTION SONT INTERDITES SANS L AUTORISATION EXPRESSE, ECRITE ET PREALABLE DE SUN MICROSYSTEMS, INC.

Cette distribution peut comprendre des composants développés par des tierces parties.

Des parties de ce produit peuvent être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, JDK, Java Naming and Directory Interface, JavaMail, JavaHelp, J2SE, iPlanet, le logo Duke, le logo Java Coffee Cup, le logo Solaris, le logo SunTone Certified et le logo Sun[tm] ONE sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Legato, le logo Legato, et Legato NetWorker sont des marques de fabrique ou des marques déposées de Legato Systems, Inc. Le logo Netscape Communications Corp est une marque de fabrique ou une marque déposée de Netscape Communications Corporation.

L'interface d'utilisation graphique OPEN LOOK et Sun(TM) a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une license non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui, en outre, se conforment aux licences écrites de Sun.

Les produits qui font l'objet de ce manuel d'entretien et les informations qu'il contient sont regis par la legislation americaine en matiere de controle des exportations et peuvent etre soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucleaires, des missiles, des armes biologiques et chimiques ou du nucleaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou reexportations vers des pays sous embargo des Etats-Unis, ou vers des entites figurant sur les listes d'exclusion d'exportation americaines, y compris, mais de maniere non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une facon directe ou indirecte, aux exportations des produits ou des services qui sont regi par la legislation americaine en matiere de controle des exportations et la liste de ressortissants specifiquement designes, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE “EN L'ETAT” ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

Contents

List of Tables

Preface

The *Java Enterprise System Upgrade Guide for UNIX* contains the information you need to upgrade Sun Java™ Enterprise System (Java ES) software in a Sun Solaris™ Operating System (Solaris OS) or Red Hat Enterprise Linux (RHEL) operating system environment. It does not cover upgrade in an HP-UX environment.

The Guide documents upgrades from Java ES 2004Q2 (Release 2), Java ES 2005Q1 (Release 3), and Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5).

This preface contains the following sections:

Who Should Use This Book

This book is intended for system administrators, or software technicians who wants to upgrade Java ES software.

This 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.

Administrative Interfaces

For most of the upgrade procedures documented in this *Upgrade Guide* there are two administrative interfaces that can be used: a graphical user interface (GUI) and a command-line interface.

In most cases, the command-line interface is used in this *Upgrade Guide* when documenting Java ES component upgrade procedures. The command-line interface can be used in scripting the upgrade of Java ES deployments so that procedures can be easily repeated when necessary.

When procedures use the Java ES installer, however, the GUI interface is described, rather than the interactive text-based interface. Java ES installer sessions can be saved in a state file that can be used to repeat procedures when necessary.

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.

Table 2 Symbol Conventions *(Continued)*

Symbol	Description	Example	Meaning
+	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`SM 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 the Java ES 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/coll/1286.2

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 5 Release Notes for UNIX http://docs.sun.com/doc/819-4893	Contains the latest information about Java ES, including known problems. In addition, components have their own release notes.
Java Enterprise System 5 Technical Overview http://docs.sun.com/doc/820-0167	Introduces the technical and conceptual foundations of Java ES. 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 ES. Presents basic concepts and principles of deployment planning and design, discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Java ES.
Java Enterprise System 5 Installation Planning Guide http://docs.sun.com/doc/819-5079	Helps you develop the implementation specifications for the hardware, operating system, and network aspects of your Java ES deployment. Describes issues such as component dependencies to address in your installation and configuration plan.
Java Enterprise System 5 Installation Guide for UNIX http://docs.sun.com/doc/819-4891	Guides you through the process of installing Java ES on the Solaris Operating System or the Linux operating system. Also shows how to configure components after installation, and verify that they function properly.

Table 4 Java Enterprise System Documentation *(Continued)*

Document	Contents
Java Enterprise System 5 Installation Reference for UNIX http://docs.sun.com/doc/819-4892	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 5 Upgrade Guide for UNIX http://docs.sun.com/doc/819-6553	Provides instructions for upgrading Java ES on the Solaris Operating System or the Linux operating environment.
Sun Java Enterprise System 5 Monitoring Guide http://docs.sun.com/doc/819-5081	Gives instructions for setting up the Monitoring Framework for each product component and using the Monitoring Console to view real-time data and set threshold alarms.
Java Enterprise System Glossary http://docs.sun.com/doc/819-3875	Defines terms that are used in Java ES documentation.

Searching Sun Product Documentation

Besides searching Sun product documentation from the `docs.sun.com` web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for "broker," type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, `java.sun.com`, `www.sun.com`, `developers.sun.com`), use `sun.com` in place of `docs.sun.com` in the search field.

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 Enterprise System 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.

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 5 in a Sun Solaris™ Operating System or Red Hat Enterprise Linux (referred to simply as Linux) operating system environment.

It contains the following sections:

Java ES 5 Components

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

Java ES components are grouped into different types, as described in the *Java Enterprise System 5 Technical Overview*, `http://docs.sun.com/doc/819-2330`:

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

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

Release 5 Product Components

Release 5 product components are listed alphabetically in the following table, along with abbreviations used in subsequent tables. For the service quality components among them, the table includes the type of service enhancement they provide.

Table 1-1 Java ES 5 Product Components

Product Component	Abbreviation	Version	Type
Access Manager	AM	7.1	System service component
Application Server	AS	8.2	System service component
Directory Proxy Server	DPS	6.0	Service quality: access component
Directory Server	DS	6.0	System service component
High Availability Session Store	HADB	4.4.3	Service quality: availability component
Java DB	JavaDB	10.2	System service component
Message Queue	MQ	3.7 UR1	System service component

Table 1-1 Java ES 5 Product Components *(Continued)*

Product Component	Abbreviation	Version	Type
Monitoring Console	MC	1.0	Service quality: administrative component
Portal Server	PS	7.1	System service component
Portal Server Secure Remote Access	PSRA	7.1	Service quality: access component
Service Registry	SR	3.1	System service component
Sun Cluster	SC	3.1 8/05	Service quality: availability component
Sun Cluster Geographic Edition	SCG	2006Q4	Service quality: availability component
Web Proxy Server	WPS	4.0.4	Service quality: access component
Web Server	WS	7.0	System service component

Release 5 Shared Components

Release 5 shared components are listed alphabetically in the following table, along with abbreviations used in subsequent tables.

Table 1-2 Java ES 5 Shared Components

Shared Component	Version	Abbreviation
Apache Commons Logging	1.0.3	ACL
Jakarta ANT Java/XML-based build tool	1.6.5	ANT
Berkeley Database	4.2.52	BDB
Common Agent Container	1.1 and 2.0	CAC
FastInfoSet	1.0.2	FIS
International Components for Unicode	3.2	ICU
Instant Messenger SDK	6.2.8	IM-SDK
Java Platform, Standard Edition	5.0 Update 7	Java SE
JavaBeans™ Activation Framework	1.0.3	JAF
Java Studio Web Application Framework	2.1.5	JATO
JavaHelp™ runtime	2.0	JavaHelp
JavaMail™ runtime	1.3.2	JavaMail

Table 1-2 Java ES 5 Shared Components *(Continued)*

Shared Component	Version	Abbreviation
Java Architecture for XML Binding runtime	2.0.3	JAXB
Java API for XML Processing	1.3.1	JAXP
Java API for XML Registries runtime	1.0.8	JAXR
Java API for XML-based Remote Procedure Call runtime	1.1.3_01	JAX-RPC
Java API for Web Services runtime	2.0	JAXWS
Java Calendar API	1.2	JCAPI
Java Dynamic Management™ Kit runtime	5.1.2	JDMK
Java Security Services (Network Security Services for Java)	4.2.4 and 3.1.11	JSS and JSS3
JavaServer Pages™ Standard Tag Library	1.0.6	JSTL
KT Search Engine	1.3.4	KTSE
LDAP C SDK	6.0	LDAP C SDK
LDAP Java SDK	4.19	LDAP J SDK
Mobile Access Core	6.2	MA Core
Netscape Portable Runtime	4.6.4	NSPR
Network Security Services	3.11.4	NSS
SOAP Runtime with Attachments API for Java	1.3	SAAJ
Simple Authentication and Security Layer	2.19	SASL
Sun Explorer Data Collector (Solaris only)	4.3.1	SEDC
Sun Java Monitoring Framework	2.0	MFWK
Sun Java Web Console	3.0.2	SJWC
Web Services Common Library	2.0	WSCL
XML Web Services Security	2.0	XWSS

Java ES Upgrade Technologies

No single system utility upgrades all Java ES components. In addition, product components and shared component upgrades have different characteristics and upgrade technologies, as described in the sections below.

Product Component Upgrades

The upgrade of Java ES product components to Release 5 is performed component-by-component, computer-by-computer, using component-specific upgrade procedures documented in this *Upgrade Guide*.

The upgrade of product components can range from major functional upgrades, which might not be compatible with the previous version of the component, to bug-fix upgrades, which are fully compatible with the previous version. Because of the dependencies between Java ES components, the nature of one upgrade can impact whether other components need to be upgraded as well.

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

- **Installation of software upgrades.** Upgraded software enhances or fixes existing software or replaces existing software. Software installation can be achieved through the application of patches to existing software packages, the selective replacement of existing packages, the installation of new packages, or a full re-installation of component software.
- **Reconfiguration.** Reconfiguration encompasses any change in configuration data, user data, or dynamic application data needed to support the upgraded software. A change in data can be additional data, a change in data format (whether in property files or database schema), or a migration of data to a new location. Sometimes reconfiguration requires that you perform a procedure and sometimes it takes place automatically. In some cases, reconfiguration also requires redeployment of component software to a web container.

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

Product Component Upgrade Approaches

The component-specific upgrade procedures used to install upgraded software and perform component reconfiguration include the following upgrade approaches:

- Using the Upgrade Capability of the Java ES Installer
- Performing a Fresh Install of the Product Component
- Running a Component-Specific Upgrade Utility
- Patching Existing Component Packages

Using the Upgrade Capability of the Java ES Installer

The Release 5 installer includes an upgrade capability that performs product component upgrade in a few special cases: Application Server, Message Queue, HADB, and Java DB. When the Java ES installer detects the previously installed release versions of these product components, it marks these components as "upgradable."

Before upgrading any of these components, the installer checks for current and previous versions of shared components. If the installer detects that a shared component required by the selected component is of a previous version or is missing, the installer upgrades all shared components currently installed and installs any missing shared components required by the selected component. In some cases (notably Application Server), the installer will also upgrade product components upon which the component being upgraded depends.

The installer removes the previous version packages, installs the Release 5 product component packages, and reconfigures, as needed, the product component being upgraded. (In the case of Application Server bundled with Solaris 9 operating system, however, the installer does not remove packages; see "Release 2 Application Server Upgrade" on page 221).

If you are using the Solaris 10 operating system zones feature, special considerations apply. See "Zone Support in the Java ES Installer" on page 59.

Performing a Fresh Install of the Product Component

Some product components are upgraded by performing a fresh install of the components using the Java ES installer. First you remove the previous version's packages and then install Release 5 in the same path, or install Release 5 in a parallel path and leave the previous version intact.

In both cases you reconfigure the product components by migrating the previous version's configuration data to the new installation, by performing a new configuration, or by doing a combination of both. For some product components, a utility is provided for reconfiguring or migrating configuration data for that component.

Running a Component-Specific Upgrade Utility

Some product components provide an upgrade utility or script for automating the upgrade of the component to Release 5. The utility generally performs both the upgrade of software packages and any reconfiguration required as part of the upgrade. For those components deployed to a web container, the utility generally redeploys the upgraded component software to the web container.

Patching Existing Component Packages

For some product components, upgrade is performed by manually patching existing software packages. While Solaris and Linux platforms employ similar technologies for managing installed software packages and tracking changes to those packages through a package registry, the differences in patching technologies between platforms impact upgrade procedures.

- **Solaris platform.** Packages are installed and removed through the Solaris `pkgadd` and `pkgrm` commands. 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 web site 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.

- **Linux platform.** Red Hat Enterprise Linux packages (RPMs) can be installed or updated through the `rpm` command. 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 Release 5 distribution, but also through the SunSolve web site at:
  http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access

  For distribution through SunSolve, RPM packages are encapsulated 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.

Upgrade Approach Used for Each Product Component

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

Table 1-3 Java ES Product Component Upgrade Approaches

Product Component	Installation of Upgraded Software	Reconfiguration
Access Manager	Replace packages: use package removal script + fresh install	Use `amconfig` and `amupgrade` scripts to re-configure and re-deploy to web container
Application Server	Replace packages: use upgrade function of Java ES installer	none, except in case of upgrade from Release 2 use `postinstall` and `asupgrade` scripts
Directory Proxy Server	Perform fresh install without replacing previous packages.	Manual reconfiguration
Directory Server	Perform fresh install without replacing previous packages.	Use `dsmig` command to migrate Directory data
High Availability Session Store	Replace packages: use upgrade function of Java ES installer or Parallel fresh install	None
Java DB	Replace packages: use upgrade function of Java ES installer	None
Message Queue	Replace packages: use upgrade function of Java ES installer or `mqupgrade` script (from Release 2)	None, except in case of upgrade from Release 2 on Linux use `mqmigrate` script
Portal Server	Replace packages: use `psupgrade` script	Use `psupgrade` script to re-configure and re-deploy to web container
Portal Server Secure Remote Access	Replace packages: use `psupgrade` script	`psupgrade` script to re-configure
Service Registry	Perform fresh install without replacing previous packages.	Manually reconfigure plus use `ant upgrade` script to re-deploy to Application Server domain
Sun Cluster	Replace packages: use `scinstall` script to replace binaries	Use `scinstall` script to migrate configuration
Sun Cluster Geo	Replace packages: use `uninstall` script + fresh Install	None
Web Proxy Server	Patch binaries	None
Web Server	Perform fresh install without replacing previous packages.	Use `wadm migrate-server` command to migrate server instance configuration

Shared Component Upgrades

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

The upgrading of shared components does not require reconfiguration of the components, nor pre- or post-upgrade procedures. In addition, shared component upgrades cannot be rolled back to their previous versions.

The large number (around 30) of Java ES shared components and the complex interactions between shared components and product components requires that all shared components within a single operating system instance be synchronized to the same Java ES release version. An operating system instance means a single computer running the Solaris 9, Solaris 10, or Red Hat Enterprise Linux operating system, or any of the virtual operating system environments (zones) on a computer running the Solaris 10 operating system.

Because of the synchronization requirement, you should not upgrade Java ES shared components one by one, but need to upgrade shared components to their Release 5 versions at the same time.

The synchronization of shared components to Release 5 is achieved using the Java ES installer. The installer synchronizes shared components when performing an upgrade of product components (see "Using the Upgrade Capability of the Java ES Installer" on page 32) or when performing a fresh install of product components. The installer also includes a synchronization function that upgrades any existing shared components and installs any missing shared components. For a fuller description of this function, see "Synchronize All Shared Components" on page 65.

The Upgrade Process

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

When you have tested the upgrade process in a staging environment, and have confidence that the upgrade is working properly, you can reproduce the process in your production environment.

The process involves the phases shown in the following table and documented in this *Upgrade Guide*. The phases apply to individual component upgrades as well as to your Java ES deployment as a whole.

Table 1-4 Phases in the Upgrade Process

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

Upgrade Plan Considerations

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

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

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

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

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

- Upgrade Dependencies
- Selective Upgrade or Upgrade Alls
- Supported Upgrade Paths and Strategies
- Multi-Instance Upgrades
- Operating System Considerations

Upgrade Dependencies

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

There are two types of upgrade dependencies:

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

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

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

Supported Upgrade Paths and Strategies

Your upgrade plan depends on the Java ES release you wish to upgrade to Release 5.

While it is possible to upgrade all previous releases of Java ES software to Java ES 5 (Release 5), the only supported upgrades are from Java ES 2005Q4 (Release 4), Java ES 2005Q1 (Release 3), and Java ES 2004Q2 (Release 2). While this *Upgrade Guide* provides strategies for upgrading from Java ES 2003Q4 (Release 1) and releases that pre-date Java ES, it does not provide procedures for performing such upgrades.

The following table describes the different upgrade paths to Release 5, their characteristics, and the upgrade strategies to be used in performing the upgrade.

Because of the differences between upgrade paths described in the table, and because product component upgrade procedures often depend on which release is being upgraded, the chapters in this *Upgrade Guide* that describe the upgrade of each product component are divided into sections: each representing a different upgrade path.

Table 1-5 Upgrade Paths to Java ES 5 (Release 5)

Product Version	Java ES Release	System Characteristics	Upgrade Strategies
2005Q4	Release 4	Java ES 5 (Release 5) supports a mixture of Release 4 and Release 5 product components on a single computer, but requires that shared components be synchronized to the same release. Interoperability between Release 4 and Release 5 product components has been tested, and known interface incompatibilities are noted in the *Java Enterprise System 5 Release Notes for UNIX*, http://docs.sun.com/doc/819-4893.	The coexistence of Release 4 and Release 5 product components provides for the possibility of selectively upgrading Release 4 product components to Release 5 on a single computer or within a deployment architecture consisting of multiple computers. If any Release 5 product component requires support of a Release 5 shared component, all shared components on the computer must be synchronized to Release 5.
2005Q1	Release 3	Similar to the Release 4 upgrade path, above. Java ES 5 (Release 5) supports a mixture of Release 3 (also Release 4) and Release 5 product components on a single computer, but requires that shared components be synonymized to the same release. Interoperability between Release 3 and Release 5 components has been tested, and known interface incompatibilities are noted in the *Java Enterprise System 5 Release Notes for UNIX*, http://docs.sun.com/doc/819-4893.	Similar to the Release 4 upgrade path, above. The coexistence of Release 3 and Release 5 components provides for the possibility of selectively upgrading Release 3 components to Release 5 on a single computer or within a deployment architecture consisting of multiple computers. If any Release 5 product component requires support of a Release 5 shared component, all shared components on the computer must be synchronized to Release 5.
2004Q2	Release 2	Contrasts with the Release 4 and Release 3 upgrade paths, above. Java ES 5 (Release 5) does *not* support a mixture of Release 2 and Release 5 components, neither product components nor shared components, on a single computer. Known interface incompatibilities exist between the release versions, and interoperability between Release 2 and Release 5 components is not certified (has not been tested).	When upgrading components from Release 2 to Release 5 on a single computer, all Release 2 components must be upgraded to Release 5. However, it is sometimes possible to mix Release 2 and Release 5 components residing on *different* computers within a deployment architecture.

Table 1-5 Upgrade Paths to Java ES 5 (Release 5) *(Continued)*

Product Version	Java ES Release	System Characteristics	Upgrade Strategies
2003Q4 and prior versions	Release 1 and pre-dating Java ES	Similar to the Release 2 upgrade path, above. Java ES 5 (Release 5) does not support a mixture of Release 2 and Release 5 components, neither product components nor shared components, on a single computer. Known interface incompatibilities exist between the release versions, and interoperability between Release 1 or prior releases and Release 5 components is not certified (has not been tested).	Java ES does not certify the direct upgrade of Release 1 or prior releases to Release 5. In some cases, however, you can 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`, and then upgrading from Release 3 to Release 5. In those cases, the upgrade roadmap for that component in this *Upgrade Guide* notes this possibility. In other cases the upgrade from Release 1 to Release 5 can be performed in the same way as the upgrade from Release 2 or Release 3 to Release 5, and in those cases, the upgrade roadmap for that component in this *Upgrade Guide* notes this possibility.

NOTE When product components issue an interim feature release (IFR) between official Java ES releases, the upgrade of the IFR is normally performed using the same procedure as for the preceding Java ES release. For example, if an IFR occurs between Release 3 and Release 4, the component would be upgraded using the procedure for upgrading from Release 3 to Release 5. When this is not the case (for example, for Portal Server and Portal Server Secure Remote Access), this *Upgrade Guide* documents the IFR-specific upgrade procedure.

Selective Upgrade or Upgrade All

The distinction between hard and soft upgrade dependencies allows for the possibility in your upgrade plan of selectively upgrading Java ES product components within a deployed system. Selective upgrade applies to upgrading from Release 3 and Release 4 to Release 5 on a single computer. Selective upgrade from Release 2 to Release 5 on a single computer is not supported.

In general, you have the choice of performing a selective upgrade or upgrading all Java ES product components on a computer:

- **Selective Upgrade.** In this approach you start with the Java ES product component you wish to upgrade to Release 5. You determine the hard upgrade dependencies for that component; those components also need to be upgraded. Repeat this process for each successive hard upgrade dependency until no further components need to be upgraded. This exercise specifies all Java ES product components that need to be upgraded.
- **Upgrade All.** In this approach you upgrade all deployed Java ES product components to Release 5. In some cases, due to the complexity of a deployment, it 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 1-6 Selective Upgrade Compared to Upgrade All

Upgrade Approach	Advantages	Disadvantages
Selective upgrade	Minimizes number of components to upgrade.	Results in inconsistent versions for all components in your deployed system
Upgrade all	Maintains a consistent version for all components in your deployed system.	Maximizes the number of components to upgrade

Selective upgrade was also supported in Java ES Release 4. It is therefore possible to have both Release 3 product components and Release 4 product components coexisting on a computer, both of which can be selectively upgraded to Release 5.

Multi-Instance Upgrades

The sequence of upgrade procedures in an upgrade plan depends on 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. Three technologies make use of redundant components in Java ES deployment architectures: load balancing (Directory Proxy Server, Web Server, Web Proxy Server, Application Server, Access Manager, and Portal Server), high availability techniques (Sun Cluster and High Availability Session Store), and Directory Server replication.

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

Redundant instances are usually deployed across multiple computers. For upgrade planning, you might need to isolate the upgrade of replicated components from other component upgrades in order to achieve minimal downtime. You perform all the pre-upgrade tasks for the replicated components on each computer before performing the rolling upgrade.

Each replication technology has configuration or reconfiguration 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.

The chapters in this *Upgrade Guide* that describe the upgrade of each product component describe how to perform multi-instance upgrades for their respective components.

Operating System Considerations

A number of operating system considerations can impact your Java ES upgrade plan, as described below.

Required Operating System Patches

Successful upgrade of a Java ES product component can require you to first patch the operating system or otherwise update the operating system to the level required by the Java ES 5 product component. However, rather than applying the specific patches or fixes needed in each case, it is preferable to bring the operating system to the level required by Java ES 5 as a whole before performing upgrades of specific product components.

- **Solaris platform.** Operating system patches are available through the SunSolve web site as a patch cluster, a collection of operating system patches that can be collectively applied. The patch clusters required to support Java ES Release 5 for Solaris 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/.` However, it is not necessary to update the Linux operating system before performing Java ES upgrades.

Dual Upgrades: Java ES and Operating System Softwared

Operating system and Java ES software can become misaligned when you attempt to upgrade either Java ES software or operating system software to a non-supported version. The relevant support matrix is shown in the following table.

Table 1-7 Java ES/Operating System Support Matrix

	Solaris			RHEL		
Java ES Release	**8**	**9**	**10**	**2.1**	**3.0**	**4.0**
2003Q4 (Release 1)	X	X		X		
2004Q2 (Release 2)	X	X		X		
2005Q1 (Release 3)	X	X	X	X	X	
2005Q5 (Release 4)	X	X	X	X	X	
5 (Release 5)		X	X		X	X

If an upgrade of Java ES software or operating system software would result in a non-supported configuration, then you have to perform a dual upgrade: one in which both Java ES and the operating system are upgraded. The following situations can require a dual upgrade:

- You upgrade the operating system to a version not supported by the installed Java ES software.

  For example, 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 also need to upgrade your Java ES Release 2 to a Java ES release version that supports the upgraded platform. In this case, it would be preferable to upgrade to Java ES 5 (Release 5).

- You upgrade Java ES to a version not supported by the existing operating system software.

  For example, Java ES 2005Q1 (Release 3) and Java ES 2005Q4 (Release 4) are supported on Solaris 8 and RHEL 2.1. If you want to upgrade Java ES to Release 5, however, which is not supported on Solaris 8 or RHEL 2.1, you must upgrade your operating system to versions supported by Java ES 5 (Release 5). In this case, it would be preferable to upgrade to Solaris 10 or RHEL 4.0.

In general, there are two approaches you can take to performing a dual upgrade:

- **Fresh operating system installation.** Install the new operating system followed by a fresh installation of Java ES Release 5, including migration of earlier version product component data (such as configuration data, runtime data, customizations, and so forth). The operating system installation can be on a new system (or Solaris 10 zone) or it can wipe out the existing file system. In the latter case, component data must first be backed up and then restored after the operating system installation.
- **In-place operating system upgrade.** Perform an operating system upgrade, leaving the existing file system in place, followed by an upgrade of Java ES product components to Release 5. For this to work, the operating system upgrade must have no impact on upgrade of the installed Java ES product components, their data, and required shared components.

If dual upgrade is not supported for any Java ES product component, that is, if neither of these approaches work, then you have to re-install and freshly configure that component after performing an operating system install or upgrade.

The following table shows the dual upgrade approach supported by each of the Java ES product components.

Table 1-8 Dual Upgrade Support for Java ES 5 Product Components

Product Component	Fresh Operating System Installation	In-Place Operating System Upgrade
Access Manager	Not supported	Not supported
Application Server	Supported on *same* computer only	Supported
Directory Proxy Server	Supported	Supported
Directory Server	Supported	Supported
High Availability Session Store	Performed in context of Application Server dual upgrade	Performed in context of Application Server dual upgrade
Java DB	Supported	Supported
Message Queue	Supported	Supported
Portal Server	Not supported	Supported
Portal Server Secure Remote Access	Not supported	Supported
Service Registry	Supported	Supported
Sun Cluster	Not supported	Supported

Table 1-8 Dual Upgrade Support for Java ES 5 Product Components *(Continued)*

Product Component	Fresh Operating System Installation	In-Place Operating System Upgrade
Sun Cluster Geographic Edition	Not supported	Performed in context of Sun Cluster dual upgrade
Web Proxy Server	Not supported	Supported
Web Server	Not supported	Supported

Operating System Upgrades

In some cases, upgrading the Solaris operating system overwrites existing Java ES shared components with earlier versions. In those cases the correct Java ES versions can be restored by upgrading Message Queue, which is bundled with the Solaris operating system, to Release 5. Upgrading Message Queue will force the upgrade of all resident shared components as well.

Solaris 10 Multizone Environments

A number of issues are involved in installing and upgrading Java ES components in a multizone environment. For a description of the benefits and limitations of deploying Java ES in Solaris 10 zones, and recommended practices for upgrading Java ES components in a multizone environment, see "Java ES 5 Upgrade and Solaris 10 Zones" on page 58.

Java ES Component Dependencies

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

This section provides information about Java ES component dependencies that impact your upgrade plan.

- Dependencies on Shared Components
- Dependencies on Product Components

Dependencies on Shared Components

Table 1-9 on page 47 shows the dependencies of Java ES 5 (Release 5) product components on Java ES shared components. The abbreviations for product components in the column headings of Table 1-9 are taken from Table 1-1 on page 28. The abbreviations for shared components are listed in Table 1-2 on page 29.

Within the matrix of Table 1-9 hard upgrade dependencies for Release 3 and Release 4 to Release 5 upgrades are marked “H,” and soft upgrade dependencies are marked “S.” For Release 2 to Release 5 upgrades, all shared component dependencies are, by definition, hard upgrade dependencies; all shared components must be upgraded from Release 2 to Release 5.

Table 1-9 Shared Component Dependencies of Java ES 5 (Release 5) Product Components

Shared Component	AM	AS	DPS	DS	DS Console	HADB	JavaDB	MQ	MC	PS	PSRA	SC	SCG	SR	WPS	WS
ANT		S							S	H	H			H		
ACL	S													H		
BDB	S															
CAC	H	S	H	H	H				S	S		S[1]	S[1]			
FIS																
ICU		S	H	H						S					S	S
IM-SDK										S						
Java SE	S	S	H	H	H	S	H	S	S	S	S	S	S	H	S	S
JAF	S	S								S	S			H		
JATO	S	S							S	S		S	S			
JavaHelp™	S	S						S	S							S
JavaMail ™	S	S								S	S			H		S
JAXB	S	S														S
JAXP	S	S								S	S			H		S
JAXR	S	S												H		S
JAX-RPC	S	S												H		S
JAXWS																S
JCAPI																
JDMK	H	S	H	H	H				S			S	S			S
JSS	S									S	S				S	S
JSTL																
KTSE										S					S	S
LDAP C SDK	H			H											S	S
LDAP J SDK	S															
MA Core	S									H	H					
MFWK	H			H					H							

Table 1-9 Shared Component Dependencies of Java ES 5 (Release 5) Product Components *(Continued)*

Shared Component	AM	AS	DPS	DS	DSConsole	HADB	JavaDB	MQ	MC	PS	PSRA	SC	SCG	SR	WPS	WS
NSPR	S	S	H	H				H	S	S	S	S	S		S	H
NSS	S	S		H				H	S	S	S	S	S		S	H
SAAJ	S	S								S	S			H		
SASL				H											S	S
SEDC												S	S			
SJWC	S	S			H				H			S	S			
WSCL	S	S												H		S
XWSS														H		

1. This dependency is specifically on Common Agent Container (CAC) version 1.1.

The dependencies shown in Table 1-9 for any product component represent both direct and indirect shared component dependencies: a product component might depend on a specific shared component (direct dependency) that, in turn, depends on one or more other shared components (indirect dependency). Figure 1-1 on page 49 illustrates interdependencies among shared components.

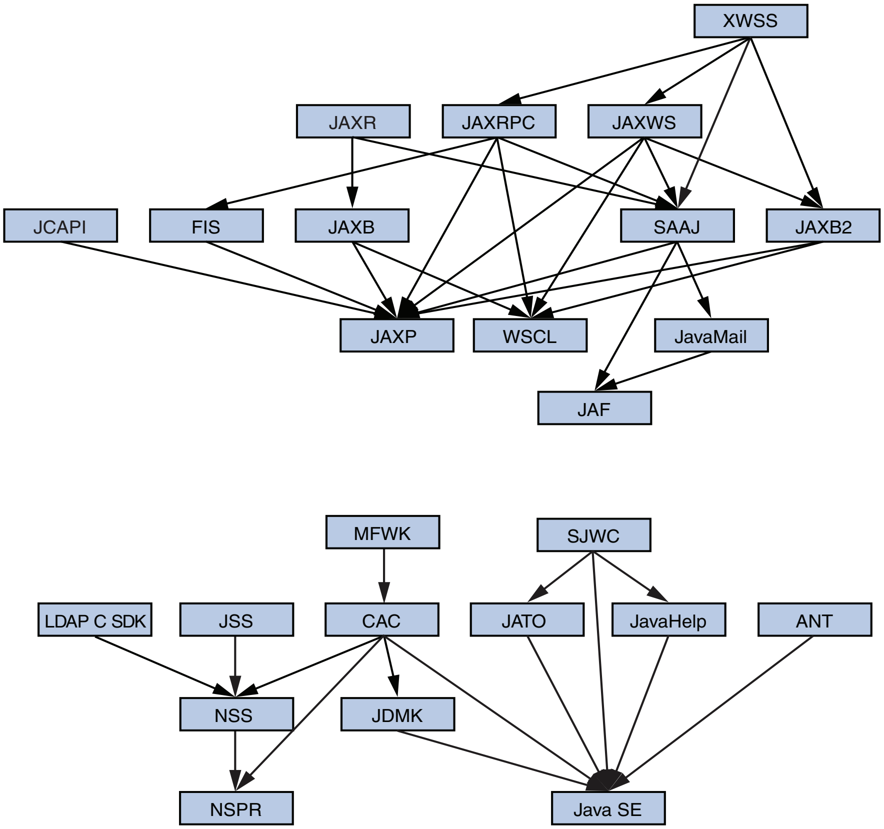
Table 1-9 shows which shared components must be upgraded when you upgrade one or more product components on a given computer.

However, because shared components must be synchronized (see "Shared Component Upgrades" on page 35), you cannot upgrade Java ES shared components one by one, but must upgrade all shared components on a computer or in an operating system instance to their Release 5 versions at the same time.

If no hard upgrade dependencies are involved, you need not upgrade shared components. However, it is a good practice to upgrade your underlying Java ES shared component base to the most current version. In fact, when product components are installed or upgraded by the Java ES installer, all shared components residing on the host computer are automatically synchronized to Release 5.

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

Figure 1-1 Shared Component Interdependencies



Dependencies on Product Components

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

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

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

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

The following table shows the dependencies and dependency relationships between the Java ES product components listed in Table 1-1 on page 28. The information can be used to determine the hard upgrade dependencies that impact your upgrade plan.

The first column alphabetically lists Release 5 product components, the second column shows other Java ES components upon which a Release 5 component has a dependency relationship, the third column provides the Java ES release versions that support the Release 5 dependency, the fourth column characterizes the dependency relationship, and the last column indicates special characteristics of the dependency, such as whether the supporting component must be local (as opposed to remote) or whether other third-party products can support the dependency.

If a product component you are upgrading to Release 5 has a dependency on Release 5 of a supporting component (as opposed to an earlier release), then the supporting component represents a hard upgrade dependency: the supporting component must also be upgraded to Release 5.

Table 1-10 Java ES Product Component Dependencies

Release 5 Product Component	Dependency[1]	Java ES Release	Nature of Dependency	Characteristics
Access Manager	Directory Server	2-5	Mandatory: Stores configuration data and enables lookup of user data	
	J2EE web container: - Application Server - Web Server	 4-5 4-5	Mandatory: Provides web container runtime services	Local only Also supported; - Weblogic[2] - WebSphere[3]
Access Manager SDK	Access Manager	3-5	Mandatory: Provides Access Manager services	
Access Manager Distr. Authentication	Access Manager	4-5	Mandatory: Provides Access Manager services	
	J2EE web container: - Application Server - Web Server	 4-5 4-5	Mandatory: Provides web container runtime services	Local only Also supported; - Weblogic[2] - WebSphere[3]
Access Manager Session Failover	Access Manager	5	Mandatory: Provides Access Manager services	
	Message Queue	4-5	Mandatory: Provides reliable asynchronous messaging	

Table 1-10 Java ES Product Component Dependencies *(Continued)*

Release 5 Product Component	Dependency[1]	Java ES Release	Nature of Dependency	Characteristics
Application Server	Message Queue	3-5	Mandatory: Provides reliable asynchronous messaging	Local only
	High Availability Session Store (HADB)	5	Mandatory: Stores session state needed to support failover between instances	Local only
	Java DB	5	Mandatory: Provides default developer database and other persistent storage.	Local only
	Web Server	3-5	Optional: Provides load balancing between instances	Local only
Directory Proxy Server	Directory Server	1-5	Co-dependency: Results in improved security and performance for directory requests. Supplies data to Directory Proxy Server	
Directory Server	Directory Proxy Server	1-5	Co-dependency: Results in improved security and performance for directory requests. Distributes load and caches data from Directory Server	
High Availability Session Store (HADB)	None			
Java DB	None			
Message Queue	Directory Server	2-5	Optional: Stores administered objects and user data	
	J2EE web container: - Application Server - Web Server	 2-5 2-5	Optional: Supports HTTP transport between client and Message Queue broker	
	Java DB	5	Optional: Stores persistent messages.	Local only
	Sun Cluster	2-5	Optional: Supports high availability	
Monitoring Cosole	None			

Table 1-10 Java ES Product Component Dependencies *(Continued)*

Release 5 Product Component	Dependency[1]	Java ES Release	Nature of Dependency	Characteristics
Portal Server	Directory Server	4-5	Mandatory: Stores and enables lookup of user profiles	
	J2EE web container: - Application Server - Web Server	 4-5 4-5	Mandatory: Provides web container runtime services	Local only
	Access Manager or Access Manager SDK	4-5	Mandatory: Provides authentication and authorization services, single sign-on	Local only (If Access Manager is remote, Access Manager SDK must be used locally)
	Portal Server Secure Remote Access	5	Optional: Provides secure remote access through the Gateway, Rewriter Proxy, and Netlet Proxy components	
	Service Registry Client	5	Mandatory: Provides libraries needed for compilation	
	Java DB	5	Mandatory: Provides support for several portlet applications	
Portal Server Secure Remote Access Gateway	Portal Server	5	Mandatory: Supports Gateway functionality	
	Access Manager or Access Manager SDK	4-5	Mandatory: Provides authentication and authorization services, single sign-on	Local only (If Access Manager is remote, Access Manager SDK must be used locally)
	Directory Server	4-5	Mandatory: Stores and enables lookup of user data	
Rewriter Proxy	Portal Server	5	Mandatory: Supports Rewriter Proxy functionality	
Netlet Proxy	Portal Server	5	Mandatory: Supports Netlet Proxy functionality	
Service Registry Deployment	Application Server	5	Mandatory: Provides container runtime services	Local only
	Java DB	5	Mandatory: Provides default database for storing services and related meta data	Local only
	Service Registry Client	5	Mandatory: Provides required client libraries	Local only

Table 1-10 Java ES Product Component Dependencies *(Continued)*

Release 5 Product Component	Dependency[1]	Java ES Release	Nature of Dependency	Characteristics
Client	None			
Sun Cluster	None			
Sun Cluster Agents	Sun Cluster	4-5	Mandatory: Provides access to Sun Cluster services	Local only
Sun Cluster Geographic Edition	Sun Cluster	4-5	Mandatory: Supports Sun Cluster Geographic Edition functionality.	Local only
Web Proxy Server	Directory Server	2-5	Optional: Provides LDAP-based authentication	
	Web Server	2-5	Co-dependency: Results in improved security and performance for HTTP requests. Supplies data to Web Proxy Server	Also supported; - Weblogic[2] - WebSphere[3]
Web Server	Directory Server	1-5	Optional: Provides LDAP-based authentication	
	Web Proxy Server	1-5	Co-dependency: Results in improved security and performance for HTTP requests. Distributes load and caches data from Web Server	

[1] For each product component, dependencies are listed in the order that they would normally be upgraded.

[2] BEA Weblogic Server

[3] IBM WebSphere Application Server

Upgrade Sequencing Guidelines

The the choice between selective upgrade or upgrade all, the impact of hard upgrade dependencies, and other factors discussed in the previous sections can all affect which Java ES components you plan to upgrade as well as the order in which you need to upgrade them. Nevertheless, a few general sequencing guidelines apply, though not in every case.

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

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

NOTES Before upgrading Java ES components, be sure to apply any required update of your operating system (see "Required Operating System Patches" on page 42).

Also check "Special Cases" on page 57 to see if any apply to your upgrade scenario.

---

1. **Shared Components** (See Chapter 2, "Upgrading Java ES Shared Components" on page 63)

   Shared components should be upgraded before the components which depend on them. In most cases shared component upgrade is handled by the Java ES installer, however in the case of Web Proxy Server and Portal Server you have to explicitly upgrade shared components.

2. **Sun Cluster software** (See Chapter 3, "Sun Cluster Software" on page 75)

   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. **Sun Cluster Geographic Edition software** (See Chapter 4, "Sun Cluster Geographic Edition" on page 87)

   Sun Cluster Geographic Edition should be upgraded after Sun Cluster software, upon which it depends. It should be upgraded before any components that use Sun Cluster services.

4. **Directory Server** (See Chapter 5, "Directory Server" on page 99)

   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.

5. **Directory Proxy Server** (See Chapter 6, "Directory Proxy Server" on page 117)

   Directory Proxy Server has a soft upgrade dependency on Directory Server and can be upgraded at any time. Some components might access Directory Server through Directory Proxy Server, however, so if Directory Proxy Server is upgraded, it should be upgraded right after Directory Server.

6. **Web Server** (See Chapter 7, “Web Server” on page 133)

   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, before upgrading Application Server.

7. Java DB (See Chapter 8, “Java DB” on page 159)

   Java DB must be upgraded before Application Server, which requires Java DB as a default database. However, Java DB is automatically upgraded by the Java ES installer when upgrading Application Server.

8. **High Availability Session Store** (See Chapter 9, “High Availability Session Store” on page 169)

   High Availability Session Store (HADB) must be upgraded before Application Server, which requires High Availability Session Store for high availability. However, HADB is automatically upgraded by the Java ES installer when upgrading Application Server.

9. **Message Queue** (See Chapter 10, “Message Queue” on page 181)

   Message Queue must be upgraded before Application Server, which requires Message Queue to be Java Enterprise Edition (Java EE) compliant. However, Message Queue is automatically upgraded by the Java ES installer when upgrading Application Server.

10. **Application Server** (See Chapter 11, “Application Server” on page 205)

    Application Server depends on Message Queue and High Availability Session Store, and if upgraded, should be upgraded after these components. Application Server can also depend on Web Server for its load balancing plug in, so if you are using that capability, Application Server should be upgraded after Web Server.

11. **Service Registry** (See Chapter 12, “Service Registry” on page 233)

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

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

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

13. **Access Manager** (See Chapter 14, "Access Manager" on page 261

    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.

14. **Portal Server** (See Chapter 15, "Portal Server" on page 309)

    Portal Server depends on many of the preceding components (Directory Server, a web container, and Access Manager), and if upgraded, should be upgraded after these components.

15. **Portal Server Secure Remote Access** (See Chapter 16, "Portal Server Secure Remote Access" on page 379)

    Portal Server Secure Remote Access, must be upgraded when Portal Server is upgraded.

Special Cases

There are a few special cases to be aware of when planning the upgrade of Java ES components to Release 5. These are described below.

Selective Upgrade: Application Server Not Upgraded

If you are performing a selective upgrade of any Java ES component to Java ES 5 on a computer that is running Release 3 or Release 4 Application Server (8.1), and you are not upgrading Application Server to Release 5, there are situations that must be addressed for Application Server to continue functioning properly:

- **JSP compilation errors**. Before performing the selective upgrade, you should first apply the Application Server patch shown in the following table.

Table 1-11 Patches[1] Needed When Application Server Is Not Upgraded to Release 5

Description	Patch ID: Solaris 9 & 10	Patch ID: Linux
Fix for Release 3 and Release 4 Application Server	119166-17 (SPARC) 119166-17 (x86)	119168-17

[1] Patch revision numbers are the minimum required. If newer revisions become available, use the newer ones instead of those shown in the table.

If you fail to apply the patch, Application Server will experience JSP compilation errors. (The patches in Table 1-11 can also be applied retroactively to fix the problem.)

- **Relocation of ANT shared component binaries on Linux.** Release 5 ANT is located in a different path from previous versions. The Application Server environment variable, specified in the *AppServer8-base*`/config/asenv.conf` file, that points to ANT must be changed from:

  ```
  AS_ANT_LIB="/opt/sun/lib"
  ```

  to:

  ```
  AS_ANT_LIB="/opt/sun/share/lib"
  ```

  and Application Server must then be restarted.

Upgrade of Portal Server Interim Feature Release (IFR) 7.0 to Java ES 5

If you are upgrading Portal Server in a Web Server environment from the Interim Feature Release (IFR) 7.0 2005Q4 to Release 5, please consult "Upgrading Portal Server from the Interim Feature Release 7.0" on page 359 for exceptions to the guidelines in "Upgrade Sequencing Guidelines" on page 54.

Java ES 5 Upgrade and Solaris 10 Zones

This section addresses issues involved in upgrading Java ES software in Solaris 10 zones and provides recommended in such an environment. The section supplements information regarding Java ES 5 and Solaris 10 zones in the *Java Enterprise System 5 Installation Planning Guide*, `http://docs.sun.com/doc/819-5079.`

It includes the following topics:

- Zone Support in the Java ES Installer
- Recommended Upgrade Practices
- Special Cases or Exceptions

Zone Support in the Java ES Installer

The Java ES 5 installer provides qualified zones support for upgrade (as well as installation) of Java ES product components and for synchronization of shared components. Policies have been implemented in the installer to help prevent problematic upgrade scenarios.

Upgrade of Product Components

As described in "Using the Upgrade Capability of the Java ES Installer" on page 32, the Java ES installer can be used to upgrade a limited number of product components and their corresponding shared components. The upgrade capability applies to global zones and all non-global zones.

However, there are three zones-related exceptions to this behavior:

- In sparse root zones, some shared components cannot be installed or upgraded because they reside in read-only directories. In such cases, the upgrade of product components is halted until such time as such shared components have been installed or upgraded in the global zone. The installer provides the following message: "The following shared components, required by the components you have selected, cannot be installed or upgraded in a sparse root zone. Please install or upgrade these shared components in the global zone before proceeding. Use the All Shared Components option."
- Both Application Server and Message Queue are bundled with the Solaris operating system. Neither of these versions can be directly upgraded in a sparse-root zone. For the details regarding these two bundled components, see "Product Component Special Cases" on page 62.
- In a global zone, if non-global zones are present, instead of upgrading all shared components currently installed and installing any missing shared components required by a selected component, the installer synchronizes *all* Java ES shared components to Release 5, whether or not they are needed by any specific product component. This allows all Release 5 shared components to be propagated to non-global zones, thus assuring that there is no intermixing of shared component versions in non-global zones.

NOTE There are a number of special cases or exceptions that might interfere with the installation or upgrade of product components in non-global zones. These are described in "Special Cases or Exceptions" on page 61.

Synchronize All Shared Components

A shared component synchronization option is provided in Release 5 to meet situations in which all shared components must be synchronized to their Release 5 versions. When the All Shared Components option is selected, the installer will upgrade all shared components currently installed and install any missing shared components, whether or not they are needed by any specific product component. This option applies to global zones and whole root zones (but not to sparse root zones).

The All Shared Components option, described in more detail in "Synchronize All Shared Components" on page 65, is needed in the following two zone-based upgrade scenarios:

- **Manually upgrading product components.** The All Shared Components option is needed to perform the shared component installation and upgrade needed when upgrading product components that *cannot* be upgraded using the Java ES installer.
- **Upgrades in a Sparse Root Zone.** Some shared components cannot be installed or upgraded in default sparse root zones. Hence, when using the Java ES installer to upgrade product components in sparse root zones, you might first be required to synchronize shared components in the global zone, depending on the shared components involved. You use the All Shared Components option in the global zone to perform the shared component installation and upgrade required in this case.

For a summary of the Java ES installer's zone behavior regarding shared components, see the information regarding Java ES 5 and Solaris 10 zones in the *Java Enterprise System 5 Installation Planning Guide*, `http://docs.sun.com/doc/819-5079.`

Recommended Upgrade Practices

In formulating an upgrade plan, you should survey any existing multizone deployments of Java ES software, keeping in mind the zones installation and administration strategies outlined in the *Java Enterprise System 5 Installation Planning Guide*, `http://docs.sun.com/doc/819-5079`. In some cases you might be required to uninstall components in one or more zones and re-install them in other zones to implement the following recommended practices:

- Avoid mixing strategies. In particular:
  - Keep your Java ES zones deployment and administration strategy as simple as possible. Do not mix whole root and sparse root deployments of Java ES components on the same computer. (Procedures and practices needed to support sparse root zone deployments can interfere with whole root zone deployments.)
  - Do not install the same Java ES product component in both the global zone and non-global zones, even if they are of different versions. (Procedures needed to upgrade a global zone installation can break the non-global zone installations.)
  - When Release 4 (or earlier) Java ES components have been installed in a whole root zone, do not upgrade Java ES components to Release 5 in the global zone. Upgrade in the global zone could result in a mixing of Release 4 and Release 5 files in the whole root zone.
- Upgrade practices:
  - If you want to upgrade all installed Release 4 product components to Release 5, synchronize all Java ES shared components in the global zone, then perform the upgrade of the desired product components in the zones where they have been installed. (Release 5 shared components are backwardly compatible.)
  - If you have Release 4 or Release 5 product components installed in a non-zones environment, and you wish to add non-global zones to the environment and install product components in the new non-global zones, you might need to uninstall components in the global zone and reinstall them in non-global zones.

Special Cases or Exceptions

There are a number of special cases, some of which arise from the fact that some Java ES shared components and some Java ES product components are bundled with Solaris 10. By virtue of this bundling, these Java ES components automatically exist in the global zone, and therefore in any non-global zone that is created from the global zone.

Product Component Special Cases

- **Message Queue.** Message Queue is bundled with Solaris 10, and, as a result, is automatically propagated when non-global zones are created (unless you have first removed Message Queue from the global zone). Message Queue cannot be installed or upgraded in a sparse root zone. When installed or upgraded in a global zone by the Java ES installer, Message Queue, unlike other product components, is, by default, propagated to non-global zones,.

- **Application Server.** Application Server is bundled with Solaris 10, and, as a result, is automatically propagated when non-global zones are created (unless you have first removed Application Server from the global zone). When propagated in this way, the bundled Application Server, which is installed in `/usr`, cannot be upgraded by the Java ES installer in a sparse root zone (by default `/usr` is read-only). To address this problem, the bundled Application Server packages must be manually removed from the global zone before installing the Release 5 Application Server in a sparse root zone. See "Solaris OS Only: Manually Remove the Application Server Packages Bundled with the Operating System" on page 222.

- **Sun Cluster.** Sun Cluster software is not supported in non-global zones.

Shared Component Special Cases

- **Sun Java Web Console (SJWC).** SJWC packages that are bundled with Solaris 10 (Update 1 and Update 2) cannot be removed by the Java ES installer. These older SJWC packages have had the `SUNW_PKG_ALLZONES` attribute set to `True`, which means the package must be identical in all zones and can only be managed by the global administrator. As a result, these packages must be manually removed in the global zone and replaced by the correct packages.

  If the Java ES installer is attempting to install a selected product component in a non-global zone and detects that SJWC needs to be upgraded, the installer will block. This will happen when installing on Solaris 10, Update 1 and 2.

  As a workaround, a special script has been developed that will remove the old packages of SJWC from the global zone and replace them with Release 5 SJWC, which has the correct zones propagation attribute value. See the *Java Enterprise System 5 Installation Guide for UNIX* for details.

- **Common Agent Container (CAC).** Version 1.1 is installed only when Sun Cluster, Sun Cluster Geographic Edition, or Sun Cluster Agents are installed. It is not installed when the All Shared Components option is selected. Only version 2.0 is installed in that case.

Chapter 2

Upgrading Java ES Shared Components

This chapter provides information on upgrading Java ES shared components to Java ES 5 (Release 5).

Each Java ES 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. They are not explicitly selected, installed, or configured during deployment of Java ES product components.

Similarly, for those product components that can be upgraded using the Java ES installer, the corresponding shared components are upgraded automatically.

However, in cases where product components are upgraded manually, as described in several chapters of this *Upgrade Guide*, the upgrade of shared components might need to be performed explicitly, using the procedures described in this chapter.

The chapter contains the following sections:

- "Shared Component Upgrade Overview" on page 64
- "Shared Component Upgrade Procedure" on page 68
- "Special Upgrade Procedures" on page 68

Be sure to read "Special Upgrade Procedures" on page 68 before performing a shared component upgrade.

Shared Component Upgrade Overview

Upgrading shared components to Java ES 5 (Release 5) 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.

This section covers the following topics:

- General Considerations
- Synchronizing Shared Components
- Synchronize All Shared Components
- Solaris 10 Zone Considerations

General Considerations

When upgrading shared components, consider the following issues:

- **Operating System Issues.** Perform any operating system upgrades, as described in "Operating System Considerations" on page 42. For all platforms except for Solaris 10 operating system, perform operating system upgrades before you upgrade shared components.
- **Sequencing Guidelines.** Review the sequencing guidelines listed in "Upgrade Sequencing Guidelines" on page 54. Typically, shared components are upgraded first. However, you should understand the entire sequence of your upgrade to Java ES Release 5 before beginning your upgrade process.

Synchronizing Shared Components

The difficulty of testing and supporting the large number (around 30) and complex interactions between Java ES shared components and Java ES product components requires that all shared components within a single operating system instance be synchronized to the same Java ES version. An operating system instance means a single computer running the Solaris 9 or Linux operating system, or in the case of the Solaris 10 operating system, it means any of the virtual operating system environments (zones) running on a single computer.

In other words, all Java ES shared components installed in an operating system instance must be of the same version. This synchronization requirement sets certain restrictions on how Java ES shared components can be installed and upgraded:

- Different versions of Java ES shared components can only reside in different operating system instances. For example, you can install Java ES Release 4 shared components in one operating system instance and Java ES Release 5 shared components in another operating system instance, but you cannot combine them in the same operating system instance.
- If any shared component in an operating system instance is upgraded or any new shared component of higher version is introduced, then all shared components in that operating system instance must also be upgraded at the same time. (Shared components are required to be backwardly compatible, so there is no problem for Release 4 product components to work with Release 5 shared components.)

For example, suppose an Release 5 product component is installed in an operating system instance in which one or more Release 4 product components reside. Because the Release 5 product component requires some number of Release 5 shared components, the synchronization requirement means that all Release 4 shared components residing in that operating system instance must be upgraded to Release 5 at the same time the Release 5 product component is installed. (This is the case even if the Release 5 product component being installed requires different shared components from those that are already installed.)

Similarly, if an Release 4 product component is upgraded to Release 5, and that upgrade requires the upgrade of some number of shared components upon which it depends, then all shared components installed in that operating system instance must be upgraded to Release 5, whether or not the particular Release 4 product component being upgraded has dependencies upon all of them or not.

Synchronize All Shared Components

The Java ES installer includes a synchronize all shared components function for situations in which all shared components must be synchronized to their Release 5 versions.

When All Shared Components is selected in the component selection page of the Java ES installer, the installer will upgrade any existing shared components and install any missing shared components, whether or not they are needed by any specific product component.

The synchronize all shared components function supports the upgrade of product components *not* explicitly upgraded by the Java ES installer (that is, all Java ES product components except Application Server, Message Queue, HADB, and Java DB). For example, the synchronize all shared components function is used to upgrade Web Proxy Server and Portal Server to Release 5.

The rationale for the synchronization of *all* shared components in this case is that the installer currently has no knowledge of which shared components need to be synchronized on a computer. Therefore this function installs or upgrades *all* Java ES shared components to the Release 5 version.

The synchronize all shared components function also supports a number of zones scenarios as described in "Solaris 10 Zone Considerations," below.

Solaris 10 Zone Considerations

Some of the limitations in the way Java ES can be deployed in a Solaris 10 multi-zone environment derive from shared component considerations.

Foremost of these considerations is that a large number of shared components cannot be installed in sparse root zones because of the read-only file systems in sparse root zones. This limitation applies to those shared components whose base directory is `/usr` (a directory that by default is shared by the global zone).

The inability to install a number of Java ES shared components in sparse root zones means that to successfully install or upgrade product components which have dependencies on such shared components into sparse root zones, the shared components must first be installed or upgraded in the global zone from which they propagate to non-global zones.

Because of the synchronization requirements regarding shared components (see "Synchronizing Shared Components" on page 64), and because the installation or upgrade of shared components in the global zone must accommodate any product component that is being installed or upgraded in the sparse root zone, it is necessary to synchronize all shared components in the global zone to their Release 5 versions. This means upgrading any existing shared components and installing any missing shared components, whether or not they might be needed by any specific product component in a sparse root zone.

This synchronization of shared components in the global zone can be performed by selecting All Shared Components in the component selection page of the Java ES installer.

When shared components are installed in and propagate from the global zone (for example, by installing a Java ES product component in the global zone), then special care must be taken to maintain synchronization of shared components in *all* zones. Otherwise it would be possible for shared components of an earlier version in a non-global zone to be mixed with Release 5 shared components that have been propagated from the global zone.

Approaches to Upgrading Shared Components

There are two approaches possible for upgrading the shared components needed to upgrade a particular Java ES product component. One is to determine all the shared components required and manually install or upgrade these to their Release 5 versions. The other is to use the Java ES installer's synchronize All Shared Components function.

However, because of the synchronization requirement regarding shared components (see "Synchronizing Shared Components" on page 64), the only practical approach is to use the synchronize All Shared Components function of the Java ES installer.

This is because it is very difficult in most cases to determine which shared components need to be installed or upgraded in any particular case, and even if this were done successfully, the manual upgrade of the required shared components is not trivial. Some shared components can be patched to their Release 5 versions, however some require replacement of previous packages with Release 5 packages. Depending on the upgrade path, some previous packages might need to be manually removed.

In the past, the difficulty of manually upgrading shared components created a significant barrier to upgrading product components. The synchronize All Shared Components function, despite the drawback of installing shared components that might not be needed, represents a significant improvement over the manual upgrading of shared components required in Java ES Release 3 and Release 4.

Shared Component Upgrade Procedure

The procedure for upgrading shared components is to use the synchronize All Shared Components function of the Java ES installer.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Shut down services or processes as indicated in "Special Upgrade Procedures" on page 68.

3. Launch the Java ES installer.

   ```
   cd Java ES Release 5 distribution/os_arch
   ./installer
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

   After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

4. Select All Shared Components in the component selection page.

5. Confirm your choice.

   All shared components will be synchronized to their Release 5 versions.

6. Exit the Java ES installer.

   See "Special Upgrade Procedures" on page 68 for any follow-up procedures.

Special Upgrade Procedures

This section provides special procedures needed for upgrading the following shared components:

- "Java SE Upgrade Procedures" on page 69
- "Common Agent Container Upgrade Procedures" on page 71

Java SE Upgrade Procedures

Java ES Release 5 is certified for Java Platform, Standard Edition (Java SE) Version 5.0 Update 9, identified here as Java SE 5.0 Update 9. (Java SE 5.0 is sometimes referred to as developer version 1.5.0).

Like other shared components, Java SE is upgraded by the Java ES installer. However, the installer upgrades to Java SE 5.0 Update 9 whether or not other shared components are being installed or upgraded. You do not have to explicitly choose All Shared Components for Java SE 5.0 Update 9 to be installed.

NOTE When upgrading Java SE you might want to first shut down any services that depend on the currently installed Java SE. This is to avoid any problems that might arise with those services. If you do not shut down services that depend on Java SE, you should reboot your system after upgrading Java SE to the Release 5 version.

When the installer detects an older version of Java SE packages or an incomplete set of packages on your computer (the complete set is: `SUNWj5rt`, `SUNWj5rtx`, `SUNWj5dev`, `SUNWj5dmo`, `SUNWj5dvx`, `SUNWj5man`, `SUNWj5cfg`, `SUNWj5dmx`), it presents you with a dialog. The dialog, which follows the component selection page, lets you choose to automatically upgrade Java SE to the Release 5 level or to bypass that automatic upgrade in favor of manually performing the upgrade.

- **Manual Upgrade.** If you choose to perform a manual upgrade of Java SE, use the following procedure:

  a. Exit the Java ES installer before installing or upgrading other components.

  b. Manually install the appropriate version of Java SE.

  c. Relaunch the Java ES installer.

     If the installer detects the correct/complete version of Java SE, then it will allow you to proceed, otherwise, it will display the previous dialog.

- **Automatic Upgrade.** If you choose to perform an automatic upgrade of Java SE, the Java ES installer upgrades Java SE to Version 5.0 Update 9. The upgrade has the following behavior:

  - Upgrade *does not* remove a previously installed major Java SE release (for example, Java SE Version 1.4.2 or Java SE Version 1.6.*x*) because other applications might depend on that version. However, the upgrade sets a symbolic link shown in the table below to reference the Release 5 version.

Table 2-1 Symbolic Link to Java SE by Platform

Platform	Symbolic Link	Location
Solaris	`/usr/jdk/entsys-j2se`	`/usr/jdk/instances/jdk1.5.0`
Linux	`/usr/jdk/entsys-j2se`	`/usr/java/jdk1.5.0_09`

You should maintain any pointers to major Java SE releases (for example, Java SE 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 Java SE.

- Upgrade *does* remove a previously installed minor Java SE release (for example, Java SE Version 5.0 Update 5) and replaces it with Java SE 5.0 Update 9.

Checking the Java SE Symbolic Link

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

Check the symbolic link (for example, on Solaris operating system) as follows:

```
ls   -l /usr/jdk/entsys-j2se
lrwxrwxrwx  1 root   other  7 Jul 7 23:18 /usr/jdk/entsys-j2se ->
   /usr/jdk/instances/jdk1.5.0
```

where `/usr/jdk/instances/jdk1.5.0` is the default location.

Verifying the Current Java SE Version

To determine which version of Java SE your Java ES installation is using, run the following command, which verifies the version of Java SE referenced by the Java SE symbolic link:

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

The outputs are shown in the following table.

Table 2-2 Java SE Version Verification Outputs

Java ES Release	Java SE Version Number	Java SE Version String
Release 2	1.4.2 Update 5	`1.4.2_05`

Table 2-2 Java SE Version Verification Outputs

Java ES Release	Java SE Version Number	Java SE Version String
Release 3	5.0 Update 1	`1.5.0_01`
Release 4	5.0 Update 4	`1.5.0_04`
Release 5	5.0 Update 9	`1.5.0_09`

Common Agent Container Upgrade Procedures

If you are upgrading the Common Agent Container shared component in preparation for upgrading Sun Cluster software (that is, patching version 1.1), follow the procedures for upgrading dependency software in "Upgrading Sun Cluster Software" in the *Sun Cluster Software Installation Guide for Solaris OS*, `http://docs.sun.com/doc/819-0420/6n2rlnncr?a=view.`

For updates to the above Guide, see "Upgrading to Sun Cluster 3.1 8/05 Software" in the *Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions*, `http://docs.sun.com/doc/819-4351`.

If you are upgrading the Common Agent Container shared component in preparation for upgrading other Java ES components (that is, upgrading to version 2.0), use the following procedure, noting the path name variables below:

Table 2-3 Common Agent Container Directory Paths

Path Name Variable	Solaris OS	Linux OS
rel4CACbase-dir	`/opt/SUNWcacao`	`/opt/sun/cacao`
rel5CAC-admin-dir	`/usr/lib/cacao`	`/opt/sun/cacao`

1. If the current installation uses custom configuration settings, capture the configuration settings using the following commands:

   rel4CAC-base-dir`/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 shows the default values. Note any nondefault settings for use in Step 4 on page 72.

2. Stop Common Agent Container processes using the following commands:

```
rel4CAC-base-dir/bin/cacaoadm stop
echo $?
```

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

```
rel4CAC-base-dir/bin/cacaoadm stop -f
```

3. Upgrade the Common Agent Container using the synchronize All Shared Components function of the Java ES installer.

See "Shared Component Upgrade Procedure" on page 68.

4. Apply any custom configuration settings previously captured in Step 1 on page 71.

```
rel5CAC-admin-dir/bin/cacaoadm set-param java-flags=Value
rel5CAC-admin-dir/bin/cacaoadm set-param jmxmp-connector-port=Value
rel5CAC-admin-dir/bin/cacaoadm set-param snmp-adaptor-port=Value
rel5CAC-admin-dir/bin/cacaoadm set-param
    snmp-adaptor-trap-port=Value
rel5CAC-admin-dir/bin/cacaoadm set-param
    commandstream-adaptor-port=Value
rel5CAC-admin-dir/bin/cacaoadm set-param retries=Value
```

5. If you have upgraded Java SE to Java SE Version 5, run the rebuild-dependencies utility:

```
rel5CAC-admin-dir/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].
```

6. Restart Common Agent Container services:

```
rel5CAC-admin-dir/bin/cacaoadm stop
rel5CAC-admin-dir/bin/cacaoadm start
```

7. Verify the upgrade of Common Agent Container:

```
rel5CAC-admin-dir/bin/cacaoadm status
rel5CAC-admin-dir/bin/cacaoadm verify-configuration
```

Chapter 3

Sun Cluster Software

This chapter describes how to upgrade Sun Cluster software to Java ES 5 (Release 5): Sun Cluster 3.1 8/05 software.

The chapter provides an overview of upgrade considerations for upgrading Sun Cluster software to Release 5.

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 5 (Release 5):

- About Release 5 Sun Cluster Software
- Sun Cluster Software Upgrade Roadmap
- Sun Cluster Data
- Sun Cluster Upgrade Strategy

About Release 5 Sun Cluster Software

Release 5 Sun Cluster software represents a minor upgrade with respect to Release 4 Sun Cluster software (see the *Sun Cluster Release Notes,* `http://docs.sun.com/doc/819-1405/6n3p13hac?a=view`). Release 5 Sun Cluster is essentially the same as Release 4, except that Release 5 does not support Solaris 8 operating system.

For changes and additions to the Sun Cluster 3.1 8/05 documentation set, see the *Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions,* `http://docs.sun.com/doc/819-4351`. Otherwise, procedures in the Sun Cluster 3.1 8/05 documentation set are valid for Release 5 Sun Cluster software.

NOTE If you require Sun Cluster 3.1 8/05 software on Solaris 8 OS, you must obtain Sun Cluster 3.1 8/05 software from the Java ES Release 4 distribution, which is located at `http://www.sun.com/software/javaenterprisesystem/previous/index.xml.`

The installation of Sun Cluster 3.1 8/05 software on Solaris 8 OS is no longer supported in the Java ES 5 installer.

To install Sun Cluster 3.1 8/05 software on Solaris 8 OS, perform the following steps:

1. Install Sun Cluster 3.1 8/05 software from the Java ES Release 4 distribution.
2. Install all required patches for Sun Cluster 3.1 8/05 software.

Sun Cluster Software Upgrade Roadmap

Table 3-1 shows the supported Sun Cluster upgrade paths to Java ES 5 (Release 5). The table applies to the Solaris operating system only.

Sun Cluster versions do not map one-to-one to Java ES releases. In the past 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 2, Release 3 and Release 4 Sun Cluster to Release 5 Sun Cluster, as shown in Table 3-1, includes the upgrade of Sun Cluster 3.1 4/04, Sun Cluster 3.1 9/04, and Sun Cluster 3.1 8/05 software to Release 5.

Table 3-1 Upgrade Paths to Java ES 5 (Release 5): Sun Cluster 3.1 8/05 Software

Java ES Release	Sun Cluster Software Version	General Approach	Reconfiguration Required
Release 4	Sun Cluster 3.1 8/05 (2005Q4)	Direct upgrade[1]: Performed using the Sun Cluster `scinstall` utility.	None.
Release 3	Sun Cluster 3.1 9/04 or Sun Cluster 3.1 8/05 (IFR)	Direct upgrade: Performed using the Sun Cluster `scinstall` utility.	Cluster configuration migrated to upgraded version automatically. No Reconfiguration is required in upgrading the IFR to Release 5.
Release 2	Sun Cluster 3.1 4/04 or Sun Cluster 3.1 9/04 (IFR)	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

[1] Upgrade from Release 4 to Release 5 is not necessary unless shared components that are used by Sun Cluster software are have been upgraded to Release 5.

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

Sun Cluster Upgrade Strategy

Your strategy for upgrading Sun Cluster generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Sun Cluster by presenting issues that might influence your Sun Cluster upgrade plan.

Compatibility Issues

Java ES Release 5 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 1-9 on page 47).
- **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.

Dual Upgrade

Dual upgrades, in which both Sun Cluster software and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in the context of a Sun Cluster nonrolling upgrade, but not as part of a rolling upgrade.

The details of the procedure can be found in the upgrade chapter of the *Sun Cluster Installation Guide*, `http://docs.sun.com/doc/819-0420/6n2rlnnd1?a=view`. Modifications for Release 5 Sun Cluster are documented in the *Sun Cluster 3.1 8/05 With Jave Enterprise System 5 Special Instructions*, `http://docs.sun.com/doc/819-4351`.

The procedure applies to upgrade of Solaris operating system from Solaris 8 or Solaris 9 to Solaris 10.

Upgrading Sun Cluster Software to Java ES Release 5

This section includes information about upgrading Sun Cluster software from Java ES 2005Q4 (Release 4), Java ES 2005Q1 (Release 3), and Java ES 2004Q2 (Release 2) to Java ES 5 (Release 5). The upgrade procedure is the same for the three Sun Cluster versions found in these Java ES releases: Sun Cluster 3.1 4/04, Sun Cluster 3.1 9/04, and Sun Cluster 3.1 8/05 software.

If you already have Sun Cluster 3.1 8/05 (Release 3 or Release 4) software installed, you do not need to upgrade to Release 5 unless shared components that are used by Sun Cluster software have been upgraded to Release 5.

The section covers the following topics:

- Introduction
- Sun Cluster Upgrade

Introduction

When upgrading Sun Cluster software to Java ES Release 5, 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 1-9 on page 47), these represent soft upgrade dependencies: Java ES Release 5 Sun Cluster software is compatible with the Release 4 versions of these components.
- **Backward Compatibility.** Release 5 Sun Cluster (framework) software is backwardly compatible with earlier cluster agents, and Release 5 Sun Cluster agent software is backwardly compatible with Java ES Release 4 components. However all nodes in a cluster must run the same version of framework and agent software.

- **Upgrade Rollback.** Rollback of the Release 5 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 and hardware architectures, 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 4 to Java ES Release 5:

- 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`. Modifications for Release 5 Sun Cluster are documented in the *Sun Cluster 3.1 8/05 With Jave Enterprise System 5 Special Instructions*, `http://docs.sun.com/doc/819-4351`.

Pre-Upgrade Tasks

Before you upgrade Sun Cluster software you should perform the following tasks:

- Verify Current Version Information
- Shut Down Sun Cluster Geographic Edition Infrastructure
- Prepare the Cluster Node for Upgrade
- Upgrade the Operating System
- Upgrade Sun Cluster Dependencies
- Obtain Required Configuration Information and Passwords

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 check the patch revision numbers to see if you have Release 4 or Release 5 software, as indicated in the following table.

Table 3-3 Sun Cluster Version Verification Outputs

Java ES Release	Sun Cluster Version Number	Patch Revision Numbers
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`	Solaris 9 sparc: 117949-15 Solaris 8 sparc: 117950-15 Solaris 9 x86: 117909-15
Release 5 (Sun Cluster 3.1 8/05)	`3.1u4`	Solaris 10 sparc: 120500-08[1] Solaris 9 sparc: 117949-23 Solaris 8 sparc: 117950-23 Solaris 10 x86: 120501-08 Solaris 9 x86: 117909-23

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5 for the different platforms. If newer revisions become available, use the newer ones instead of those shown in the table.

Shut Down Sun Cluster Geographic Edition Infrastructure

If you are upgrading a cluster that runs Sun Cluster Geographic Edition software, you must first shut down the Sun Cluster Geographic Edition infrastructure and perform other steps to prepare the cluster for upgrade. Follow the upgrade procedures in the Sun Cluster Geographic Edition *Installation Guide*, http://docs.sun.com/doc/819-8004/6n9tmd19d?=view. These procedures include steps to upgrade Sun Cluster software at the appropriate stage of the upgrade.

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 Release 5. Upgrade of Release 4 shared components upon which Sun Cluster software depends is optional and can be performed as described in Chapter 2, "Upgrading Java ES Shared Components." (Upgrade of Release 2 shared components to Release 5 is mandatory.)

NOTE If Java ES shared components have been synchronized to Release 5, then Sun Cluster software must be upgraded to Release 5. This is because Release 4 Sun Cluster is not compatible with the Release 5 Sun Java Web Console shared component.

If Sun Java Web Console has been upgraded to Release 5 (Ver 3.0), then you should apply the latest Sun Cluster patches or upgrade Sun Cluster as described in this chapter.

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 5 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 and 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`. Modifications for Release 5 Sun Cluster are documented in the *Sun Cluster 3.1 8/05 With Java Enterprise System 5 Special Instructions,* `http://docs.sun.com/doc/819-4351`.

1. Boot into non-cluster mode.
2. Log in as root or become superuser.

   ```
   su -
   ```

3. Change to the following directory on the Java ES Release 5 distribution:

   cd /*os_arch*/Product/sun_cluster/*os-ver*/Tools

   where *os_arch* matches your platform, such as `Solaris_sparc`, and *os-ver* is `Solaris 9` or `Solaris 10`.

4. Run the `scinstall` utility.

   ```
   ./scinstall
   ```

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

5. 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 85).

6. 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 3.0-3.1 Release Notes Supplement*, `http://docs.sun.com/app/docs/doc/816-3381/6m9lratq9?a=view#gcpom.`

7. Reboot the node into the cluster.

Verifying the Upgrade

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

1. Enter 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 check the patch revision numbers to see if you have Release 4 or Release 5 software, as indicated in Table 3-3 on page 82.

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

Sun Cluster Geographic Edition

This chapter describes how to upgrade Sun Cluster Geographic Edition to Java ES 5 (Release 5): Sun Cluster Geographic Edition 3.1 2006Q4.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on only the Solaris operating system:

- "Overview of Sun Cluster Geographic Edition Upgrades" on page 88
- "Upgrading Sun Cluster Geographic Edition from Version 3.1 8/05" on page 91

NOTE File locations in this chapter are specified with respect to a directory path referred to as *SunClusterGeo-base*. At least part of this path might have been specified as an installation directory when Sun Cluster Geographic Edition was installed. If not, the Geographic Edition installer assigned a default value.

The default values of this directory path is shown in the following table.

Table 4-1 Sun Cluster Geographic Edition Directory Paths

Path Name Variable	Solaris OS	Linux OS
SunClusterGeo-base	`/opt/SUNWscgeo`	Not Applicable

Overview of Sun Cluster Geographic Edition Upgrades

This section describes the following general aspects of Sun Cluster Geographic Edition that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5
- Java ES Release 5 Upgrade Roadmap
- Sun Cluster Geographic Edition Data
- Sun Cluster Geographic Edition Upgrade Strategy

About Java ES Release 5

Java ES Release 5 Sun Cluster Geographic Edition is the first release to be delivered as a Java ES component; Sun Cluster Geographic Edition 3.1 was first released as a standalone product.

Release 5 Sun Cluster Geographic Edition represents a minor release with respect to Sun Cluster Geographic Edition 3.1 8/05. It is the first release to support the Solaris x86 platform. Release 5 Sun Cluster Geographic Edition also includes some selected bug fixes, qualifications for various hardware and software components, and support for additional data replication products.

Java ES Release 5 Upgrade Roadmap

Table 4-2 shows the supported Sun Cluster Geographic Edition upgrade paths to Java ES Release 5. The table applies to the Solaris operating system only.

Table 4-2 Upgrade Paths to Java ES 5 (Release 5): Sun Cluster Geographic Edition 3.1 2006Q4

Java ES Release	Sun Cluster Geographic Edition Version	General Approach	Reconfiguration Required
Pre-dates Java ES releases	Sun Cluster Geographic Edition 3.1 8/05	Direct upgrade: Replace Sun Cluster Geographic Edition 3.1 8/05 with a fresh install	None

Sun Cluster Geographic Edition Data

The following table shows the type of data that can be impacted through an upgrade of Sun Cluster Geographic Edition software.

Table 4-3 Sun Cluster Geographic Edition Data Usage

Type of Data	Location	Usage
Sun Cluster Geographic Edition 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 Geographic Edition operations.

Sun Cluster Geographic Edition Upgrade Strategy

Your strategy for upgrading Sun Cluster Geographic Edition generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Sun Cluster Geographic Edition by presenting issues that might influence your Sun Cluster Geographic Edition upgrade plan.

Compatibility Issues

Both clusters in a partnership have to run the same version of Sun Cluster Geographic Edition. Hence, Release 5 Sun Cluster Geographic Edition cannot be mixed in a partnership with Sun Cluster Geographic Edition 3.1 8/05 (there is no backward compatibility). Release 5 Sun Cluster Geographic Edition includes additional configuration data which can't be read by Sun Cluster Geographic Edition 3.1 8/05.

Dependencies

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

Sun Cluster Geographic Edition has dependencies on the following Java ES components:

- **Shared components.** Sun Cluster Geographic Edition has dependencies on specific Java ES shared components (see Table 1-9 on page 47).
- **Sun Cluster.** Sun Cluster Geographic Edition has a mandatory dependency on Sun Cluster, which provides base functionality.

Dual Upgrade

Dual upgrades, in which both Sun Cluster Geographic Edition software and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) do not apply to Sun Cluster Geographic Edition.

Because Sun Cluster Geographic Edition has a hard upgrade dependency on Sun Cluster, the operating system upgrade is performed in the context of the Sun Cluster upgrade. See "Dual Upgrade" on page 79 for information about Sun Cluster dual upgrades.

Upgrading Sun Cluster Geographic Edition from Version 3.1 8/05

This section includes information about upgrading Sun Cluster Geographic Edition from version 3.1 8/05 to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Version 3.1 8/05 Sun Cluster Geographic Edition Upgrade

Introduction

When upgrading Sun Cluster Geographic Edition 3.1 8/05 to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is achieved by uninstalling Version 3.1 8/05 and performing a fresh install of Release 5 Sun Cluster Geographic Edition. Configuration data is retained.
- **Upgrade Dependencies.** Sun Cluster Geographic Edition has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Sun Cluster Geographic Edition. In particular, Sun Cluster Geographic Edition has a hard upgrade dependency on Common Agent Container, version 1.1. Sun Cluster Geographic Edition also has a hard upgrade dependency on Sun Cluster: Release 4 Sun Cluster (3.1 8/05) must be upgraded to Release 5 to support Release 5 Sun Cluster Geographic Edition.
- **Backward Compatibility.** Release 5 Sun Cluster Geographic Edition is not compatible with Sun Cluster Geographic Edition 3.1 8/05. All clusters in a partnership must be upgraded to Release 5.
- **Upgrade Rollback.** Rollback of the Release 5 upgrade of Sun Cluster Geographic Edition software to version 3.1 8/05 is not supported.
- **Platform Issues.** The approach for upgrading Sun Cluster Geographic Edition software is the same on all Solaris platforms and hardware architectures, however Sun Cluster Geographic Edition software is not supported on Linux platforms.

Version 3.1 8/05 Sun Cluster Geographic Edition Upgrade

This section describes how to perform an upgrade of Sun Cluster Geographic Edition from version 3.1 8/05 to Java ES Release 5. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading 3.1 8/05 Sun Cluster Geographic Edition
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Sun Cluster Geographic Edition software you should perform the following tasks:

- Verify Current Version Information
- Prepare a Cluster for an Upgrade
- Upgrade Sun Cluster Geographic Edition Dependencies
- Back Up Sun Cluster Geographic Edition Data
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Sun Cluster Geographic Edition using the following command:

```
/usr/cluster/bin/geoadm --version
```

The version string follows the copyright text, and is shown in the following table:

Table 4-4 Sun Cluster Geographic Edition Version Verification Outputs

Java ES Release	Sun Cluster Geographic Edition Version Number
Version 3.1 8/05	`1.0`
Release 5	`1.1`

Prepare a Cluster for an Upgrade

Perform the following steps on all clusters that have a partnership with the cluster you are upgrading. The procedure removes the Sun Cluster Geographic Edition layer from production.

1. Ensure that the cluster is functioning properly.

   To view the current status of the cluster, run the following command from any node.

   ```
   scstat
   ```

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Remove all application resource groups from protection groups.

   Highly available applications do not have downtime during the Sun Cluster Geographic Edition upgrade.

   geopg remove-resource-group *resourcegroup protectiongroupname*

4. Perform Step 1 through Step 3 on all clusters that have a partnership with this cluster.

5. Stop all protection groups that are active on the cluster.

   geopg stop *protectiongroupname* -e local | global

6. Remove the ICRM plug-in from all the heartbeats on both partner clusters.

   geohp remove-plugin *icrmplugin heartbeatname*

Upgrade Sun Cluster Geographic Edition 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 5. Sun Cluster Geographic Edition has hard upgrade dependencies on a number of shared components and on Sun Cluster.

When upgrading Sun Cluster Geographic Edition dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Sun Cluster Geographic Edition. Upgrade of shared components is normally achieved automatically by the Java ES installer.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Sun Cluster Geographic Edition are upgraded automatically by the Java ES installer when you perform an upgrade of Sun Cluster Geographic Edition to Release 5.

   NOTE If Java ES shared components are upgraded to Release 5, then Sun Cluster Geographic Edition software must also be upgraded to Release 5. This is because Sun Cluster Geographic Edition 3.1 8/05 is not compatible with Release 5 Sun Java Web Console.

2. **Sun Cluster.** Instructions for upgrading Sun Cluster to Release 5 are provided in Chapter 3, "Sun Cluster Software" on page 75.

Back Up Sun Cluster Geographic Edition Data

Sun Cluster Geographic Edition stores all data in the Cluster Configuration Repository, so 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 3.1 8/05 Sun Cluster Geographic Edition

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

Upgrade Considerations

The upgrade of Sun Cluster Geographic Edition software to Java ES Release 5 should be possible without disturbing running applications. You can upgrade Sun Cluster Geographic Edition software on a running cluster without disruption; the cluster remains in production with services running. Similarly, you can also apply Sun Cluster Geographic Edition patches without downtime. Configuration data is retained across the upgrade process.

Sun Cluster Geographic Edition software must be upgraded on all nodes of all clusters that have a partnership with the cluster you are upgrading.

If you want upgrade the Solaris operating system during the Sun Cluster Geographic Edition software upgrade process, you must remove the Sun Cluster Geographic Edition packages before you upgrade the Solaris operating system.

Upgrade Procedure

The procedure documented below applies to Sun Cluster Geographic Edition instances residing locally on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Shut down Sun Cluster Geographic Edition 3.1 8/05.

   ```
   /usr/cluster/bin/geoadm stop
   ```

   To leave the underlying applications running while shutting down Sun Cluster Geographic Edition, see the Sun Cluster Geographic Edition *System Administration Guide*, http://docs.sun.com/doc/819-8003.

3. Uninstall Sun Cluster Geographic Edition 3.1 8/05.

   SunClusterGeo-base`/install/uninstall/uninstaller`

   Use the Sun Cluster Geographic Edition installer in uninstall mode, as documented in the Sun Cluster Geographic Edition *Installation Guide*, http://docs.sun.com/doc/819-8004.

4. If Sun Cluster is not running start it up.

   Sun Cluster must be running to install Sun Cluster Geographic Edition. You can check it using the `scstat` command.

5. Perform a fresh install of Release 5 Sun Cluster Geographic Edition using the Java ES installer.

   a. Launch the Java ES installer on the computer hosting Release 4 Access Manager.

      `cd` *Java ES Release 5 distribution*`/`*os_arch*
      `./installer`

      where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

      After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of “upgradable.”)

   b. Select Sun Cluster Geographic Edition from the component selection page.

c. Choose to Configure Now or Configure Later.

   It makes no difference whether you choose to configure now or to configure later because no Reconfiguration is required.

d. If needed, select the option to install localized packages.

e. Exit the Java ES installer when installation is complete.

NOTE If you are upgrading Sun Cluster Geographic Edition software on Solaris 8 OS, you must use the pkgadd command to install the software from the Java ES Release 5 distribution. The Java ES installer does not support the installation of Sun Cluster Geographic Edition software on Solaris 8 OS.

6. Install all the required Sun Cluster Geographic Edition patches.

   For instructions, see the Sun Cluster Geographic Edition *Installation Guide*, http://docs.sun.com/doc/819-8004.

7. Perform Step 1 on page 95 through Step 6 on all clusters that have a partnership with this cluster.

8. Re-start Release 5 Sun Cluster Geographic Edition software.

```
/usr/cluster/bin/geoadm start
```

   If the cluster is in a partnership, all nodes on both partners must be upgraded before Sun Cluster Geographic Edition software is started up.

   For more information, consult the *Installation Guide* referenced above and the Sun Cluster Geographic Edition *System Administration Guide*, http://docs.sun.com/doc/819-8003.

9. Add all application resource groups you removed when preparing for a cluster upgrade as described in "Prepare a Cluster for an Upgrade" on page 93.

```
geopg add-resource-group resourcegroup protectiongroupname
```

10. Start all the protection groups you have added.

```
geopg start protectiongroupname -e local | global [-n]
```

Verifying the Upgrade

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

1. Run the following command:

   ```
   /usr/cluster/bin/geoadm --version
   ```

   See Table 4-4 on page 92 for output values.

2. If applicable, after upgrading both partners, run the following commands on one cluster node of each partner.

   ```
   /usr/cluster/bin/geoadm start
   /usr/cluster/bin/geoadm show
   ```

   The command will show whether Sun Cluster Geographic Edition software is active on that node.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure" on page 95.

Rolling Back the Upgrade

Rollback of Sun Cluster Geographic Edition software is not supported.

However, you can remove Release 5 Sun Cluster Geographic Edition using the uninstall function of the Java ES installer and then use the version 3.1 8/05 installer to re-install version 3.1 8/05. Because Release 5 Sun Cluster Geographic Edition writes data which version 3.1 8/05 cannot read, any such rollback would have to be performed before Release 5 had been started, that is before you start the upgraded Sun Cluster Geographic Edition.software, as described in "Upgrade Procedure" on page 95.

Chapter 5

Directory Server

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

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. 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 *serverRoot* (Directory Server 5.*x*) and *DirServer-base* (Directory Server 6.0). At least part of these paths might have been specified as installation directories when Directory 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.

Table 5-1 Directory Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
serverRoot (Directory Server 5.*x*)	`/var/opt/mps/serverroot`	`/var/opt/sun/directory-server`

Table 5-1 Directory Server Directory Paths *(Continued)*

Path Name Variable	Solaris OS	Linux OS
DirServer-base (Directory Server 6.0)	`/opt/SUNWdsee`	`/opt/sun`

Overview of Directory Server Upgrades

This section describes the following general aspects of Directory Server that impacts upgrading to Java ES 5 (Release 5):

- About Java ES Release 5
- Java ES Release 5 Upgrade Roadmap
- Directory Server Data
- Directory Server Upgrade Strategy

About Java ES Release 5

Java ES Release 5 Directory Server represents a major release, with a variety of new features and improvements. See the *Directory Server Enterprise Edition 6 Release Notes*, `http://docs.sun.com/doc/819-0991` for details.

Java ES Release 5 Upgrade Roadmap

Table 5-2 shows the supported Directory Server upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 5-2 Upgrade Paths to Java ES 5 (Release 5): Directory Server 6.0

Java ES Release	Directory Server Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Directory Server 5.2 2005Q4	Direct upgrade: Fresh install and migration of all data.	Configuration data migrated from previous version to newly installed Directory Server
	Sun Java System Administration Server 5.2 2005Q4	Administration Server functionality replaced by Directory Service Control Center and Directory Server EE command-line utilities.	
Release 3	Sun Java System Directory Server 5 2005Q1	Direct upgrade: Fresh install and migration of all data.	Configuration data migrated from previous version to newly installed Directory Server
	Sun Java System Administration Server 5 2005Q1	Administration Server functionality replaced by Directory Service Control Center and Directory Server EE command-line utilities.	

Table 5-2 Upgrade Paths to Java ES 5 (Release 5): Directory Server 6.0

Java ES Release	Directory Server Version	General Approach	Reconfiguration Required
Release 2	Sun Java System Directory Server 5.2 2004Q2 Sun Java System Administration Server 5.2 2004Q2	Direct upgrade: Fresh install and migration of all data. Administration Server functionality replaced by Directory Service Control Center and Directory Server EE command-line utilities.	Configuration data migrated from previous version to newly installed Directory Server
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.	Configuration data migrated from previous version to newly installed Directory Server
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.	Configuration data migrated from previous version to newly installed Directory Server
	Sun ONE Directory Server 5.1 Sun ONE Administration Server 5.1	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 5.	Refer to the *Java Enterprise System 2005Q1 Upgrade and Migration Guide*, `http://docs.sun.com/doc/819-0062`.

Directory Server Data

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

Directory Server 6.0 no longer stores configuration data in a configuration directory. Configuration is performed using the Directory Service Control Center (or the Directory Server EE command-line utilities), and should be accessed through this interface. Directory Service Control Center stores configuration data in its own local Directory Server instance.

The following table shows the type of data that is impacted by an upgrade of Directory Server software to Release 5.

Table 5-3 Directory Server Data Usage

Type of Data	Location	Usage
Directory Server configuration data	Directory Server 5.1 and 5.2: configuration directory Directory Server 6.0: accessed through Directory Service Control Center and Directory Server EE command-line utilities	Configuration of Directory Server instances
Directory Server schema		Define structure and semantics of data in the directory
Security data	Directory Server 5.1 and 5.2: SSL configured through Directory Server Console. Directory Server 6.0: SSL configured through Directory Service Control Center and Directory Server EE command-line utilities.	Server certificates.
User data	Directory Server	Support applications with user-specific configuration data and user profiles

Directory Server Upgrade Strategy

Your strategy for upgrading Directory Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Directory Server by presenting issues that might influence your Directory Server upgrade plan.

Compatibility Issues

Java ES Release 5 Directory Server does not introduce new public interfaces and is therefore backwardly compatible with earlier versions; it supports all components supported by Release 4 Directory Server and earlier versions.

However, Release 5 introduces changes to private administrative interfaces. The Release 5 interfaces are incompatible with earlier releases of Directory Server. In particular, the Administration Server, used to configure earlier Directory Server instances, has been replaced by the Directory Service Control Center and Directory

Server EE command-line utilities, and the o=NetscapeRoot directory suffix for storing Directory Server configuration information has been eliminated. Details can be found in the *Directory Server Enterprise Edition 6 Migration Guide,* http://docs.sun.com/doc/819-0994.

Dependencies

Dependencies on other Java ES components can, in general, impact the procedure for upgrading Directory Server software.

Directory Server has dependencies on the following Java ES components:

- **Shared components.** Directory Server has dependencies on specific Java ES shared components (see Table 1-9 on page 47). Directory Server upgrades might depend upon upgraded versions of these shared components.
- **Directory Proxy Server.** Directory Server has a co-dependency on Directory Proxy Server for providing improved security and performance for LDAP requests.

Dual Upgrade

Dual upgrades, in which both Directory Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up the existing Directory Server data.

   See "Create Directory Server Image (Optional)" on page 108 regarding Directory Server 5.*x* information.

2. Install the new operating system.

   The operating system installation can be on a new system (or a Solaris 10 zone) or it can wipe out the existing file system.

3. Restore the Directory Server data that was backed up in Step 1.

4. Install Release 5 Directory Server.

5. Create a Release 5 Directory Server instance and migrate directory data to the new instance.

   See the relevant steps in the procedure for "Upgrading Release 4 Directory Server" on page 108.

In-place Operating System Upgrade

1. Back up the existing Directory Server data.

   See "Create Directory Server Image (Optional)" on page 108 regarding Directory Server 5.*x* information.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Directory Server.

   See the relevant section of this chapter, depending on upgrade path.

Upgrading Directory Server from Java ES Release 4

This section includes information about upgrading Directory Server from Java ES 2005 Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Directory Server Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is achieved by performing a fresh install of Release 5 Directory Server and then using migration tools to re-create the previous Directory Server instances in new, distinct Release 5 Directory Server instances.
- **Upgrade Dependencies.** Directory Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Directory Server.
- **Backward Compatibility.** Release 5 Directory Server is not backwardly compatible with the Release 4 version, as described in "Compatibility Issues" on page 103. However, the migration tools make it possible to migrate the o=NescapeRoot suffix if you continue to maintain a set of Directory Server instances relying on the Directory Server 5.*x* administration framework.
- **Upgrade Rollback.** A rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade to Release 5.
- **Platform Issues.** The general approach for upgrading Directory Server is the same on both Solaris and Linux operating systems.

Release 4 Directory Server Upgrade

This section describes how to perform an upgrade of Directory Server from Java ES Release 4 to Java ES Release 5 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
- Upgrading Release 4 Directory Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Directory Server software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Directory Server Dependencies
- Obtain Required Configuration Information and Passwords
- Create Directory Server Image (Optional)

Verify Current Version Information

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

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

Table 5-4 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
Release 5	Sun Java(TM) System Directory Server/6.0

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 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 5. However, all shared components required by Directory Server are upgraded automatically when you perform an upgrade of Directory Server to Release 5.

Obtain Required Configuration Information and Passwords

You should know the Directory Server administrator user ID and password for your currently installed version. Other configuration information is preserved through the upgrade process.

Create Directory Server Image (Optional)

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

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

Upgrading Release 4 Directory Server

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

Upgrade Considerations

The upgrade of Directory Server software to Java ES Release 5 takes into account the following considerations:

- Any Java ES components using a Directory Server instance (such as Access Manager or Portal Server, or Sun Java Communications Suite components) should be shut down and re-configured, if needed, to access the corresponding new Release 5 instance.

- 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), you only have to upgrade the Directory Server image once; however, you have to separately migrate the data for each of the instances.
- In many Release 4 Directory Server deployment architectures the configuration directory is a separate Directory Server instance. These instances do not need to be upgraded because the configuration directory has been deprecated in Release 5. On the other hand, the upgrade might entail the deployment of the Release 5 Directory Server administrative console (the Directory Service Control Center) to a separate computer from which you remotely manage Directory Server instances.
- A command line tool is provided with Directory Server, which helps automate the migration of schema, configuration, security and user data. The migration tool allows a step by step migration of these different data. Most upgrade scenarios benefit from automated migration of at least some of the data.

Upgrade Procedure

The procedure documented below applies to Directory Server instances residing locally on the computer where the upgrade is taking place, or in the case where instances are moving to another computer, all instances that will run on the target computer.

1. Log in as root or become superuser.

   `su -`

2. Shut down the Release 4 Directory Server (5.2) instances.

   serverRoot`/slapd-`*instanceName*`/stop-slapd`

   Check that the error log (*serverRoot*`/slapd-`*hostName*`/logs/errors`) reports a clean shutdown:

   ```
   [23/Jan/2006:15:56:47 +0100] - All database threads now stopped

   [23/Jan/2006:15:56:50 +0100] - slapd stopped.
   ```

3. Ensure that the host computer for Release 5 Directory Server has sufficient disk space.

   The basic calculation is as follows:
   2 * (*space for existing server*) + (*space for LDIF files*)

   There is unfortunately no tool allowing to anticipate the size of an LDIF file created from an exported database. The size will depend upon the number of data entries, their internal representation, the number of indexes, and so forth.

4. For remote install of Release 5, create a Release 4 image and transfer it to the remote computer.

   See "Create Directory Server Image (Optional)" on page 108.

5. Make sure you have upgraded any Java ES components upon which Directory Server has hard upgrade dependencies (see "Upgrade Directory Server Dependencies" on page 108).

6. Perform a fresh install of Release 5 Directory Server.

   Perform the following steps:

   a. Launch the Java ES installer.

      ```
      cd Java ES Release 5 distribution/os_arch

      ./installer
      ```

      where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

      After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

   b. Select the Directory Server subcomponent of Directory Server Enterprise Edition.

      You will also need to install the administrative subcomponents (Directory Service Control Panel or command line utilities) you wish to use.

   c. Specify an installation path different from that of any existing Release 4 Directory Server.

d. Choose to Configure Now or Configure Later.

It does not matter whether you choose to Configure Now or to Configure Later because there is really no configuration required for Directory Server. However, if you choose to Configure Now, do *not* opt to create a new instance.

e. Confirm your installation choices.

Directory Server packages will be upgraded and an upgrade summary displayed.

f. Exit the Java ES installer.

7. Create a Directory Server instance.

DirServer-base`/ds6/bin/dsadm create` *instancePath*

where *instancePath* is the full path to the Directory Server instance.

For information on creating a Directory Server instance, see the *Directory Server Enterprise Edition 6 Administration Guide*, `http://docs.sun.com/doc/819-0995`.

If you fail to create a new instance, a new instance will automatically be created for you when you migrate data with the `dsmig` command (Step 8).

If the `dsadm` command fails on the Solaris 10 platform, set the library path to null when running the command:

`LD_LIBRARY_PATH= ./dsadm create` *instancePath*

8. Migrate Release 4 data to the Release 5 Directory Server instance.

Use the *DirServer-base*`/ds6/bin/dsmig` commands.

The `dsmig` commands adapt the Release 4 data to the Release 5 format and write it to the appropriate locations. For example, a typical migration on a single computer with one Directory Server instance might look like this:

```
dsmig migrate-schema -v old_instancePath new_instancePath

dsmig migrate-config -v old_instancePath new_instancePath

dsmig migrate-security -v old_instancePath new_instancePath

dsmig migrate-data -v old_instancePath new_instancePath
```

NOTES

- If the `dsmig migrate-config` command fails on the Solaris 10 platform, set the library path to null when running the command:

  ```
  LD_LIBRARY_PATH= ./dsmig migrate-config ...
  ```

- If the Directory Server instance you're migrating is storing configuration data for other Java ES components, for example for the Sun Java Communications Suite Messaging Server component, it might be required that you migrate a specific part of the directory information tree named `o=netscaperoot`. This root suffix is not migrated by default. To migrate `o=netscaperoot`, use the `-N` option of the `dsmig migrate-config` and `dsmig migrate-data` commands. For example:

  ```
  dsmig migrate-config -v old_instancePath
      new_instancePath -N
  ```

- If you are migrating from an instance on a 32-bit architecture to one on a 64-bit architecture, you cannot use the `dsmig migrate-data` command (automatic migration tool). You have to migrate the data manually, as documented in the *Migration Guide* referenced below. However you can still perform automatic migration of schema, configuration, and security data.

- In some cases, when starting Directory Server after migrating directory data, new Release 5 error checking detects circular definitions in Directory Server group entries. These circular definitions are functionally benign, but can result in a large number of errors being logged into the error file.

For details of the migration process, the `dsmig` commands, and manual migration, see the *Directory Server Enterprise Edition 6 Migration Guide*, `http://docs.sun.com/doc/819-0994`.

Verifying the Upgrade

You can verify successful upgrade of Directory Server as follows.

1. Start the new Directory Server instance:

   DirServer-base`/ds6/bin/dsadm -V`

   See Table 5-4 on page 107 for output values.

2. Check the startup messages in the Directory Server error log:

   instancePath`/logs/errors`

NOTE At startup, Release 5 Directory Server now detects circular definitions. These circular definitions are functionally benign, but can result in a large number of errors when you upgrade from a previous version which contains such circular definitions.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure" on page 109, except that all Java ES components dependent on Directory Server need to be re-configured to point to the new Directory Server instances.

Rolling Back the Upgrade

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

Multiple Instance Upgrades

The procedures in "Release 4 Directory Server Upgrade" on page 107 do not explicitly deal with deployment architectures in which Directory Server is replicated for availability or scalability. These architectures might include Directory Server replication or the deployment of Directory Server as a data service in a Sun Cluster environment.

Rolling Upgrades of Directory Server Replicates

Multiple instances of Directory Server on different computer systems, such as used in multi-master replication deployment architectures, can be sequentially upgraded one instance at a time. After first synchronizing all Directory Server masters, you upgrade each instance on its respective host computer 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

Information regarding upgrade and roll back of Directory Server as a data service in a Sun Cluster environment is currently under development.

Upgrading Directory Server from Java ES Release 3

The procedure for upgrading Java ES 2003Q1 (Release 3) Directory Server to Release 5 is the same as that for upgrading Release 4 Directory Server to Release 5.

To upgrade Release 3 Directory Server to Release 5, use the instructions in "Upgrading Directory Server from Java ES Release 4" on page 106, except substitute Release 3 wherever Release 4 is referenced.

Upgrading Directory Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Directory Server to Release 5 is the same as that for upgrading Release 4 Directory Server to Release 5, with the exception that the pre-upgrade tasks should include the upgrading to Release 5 of all shared components (see Table 1-9 on page 47).

Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

To upgrade Release 2 Directory Server to Release 5, use the instructions in "Upgrading Directory Server from Java ES Release 4" on page 106, except substitute Release 2 wherever Release 4 is referenced.

NOTE If you are upgrading from Release 2 Directory Server on the Linux platform, then you will have to perform a dual upgrade, in which both Directory Server *and* the operating system are upgraded (Release 5 Directory Server is not supported on RHEL 2.1). See "Dual Upgrade" on page 104 for more information.

Chapter 6

Directory Proxy Server

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

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. 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 *serverRoot* (Directory Proxy Server 5.*x*) and *DirServer-base* (Directory Proxy Server 6.0). At least part of these paths might have been specified as installation directories when Directory Proxy Server was installed. If not, the Java ES installer assigned a default value.

The default values of these directory paths are shown in the following table.

Table 6-1 Directory Proxy Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
serverRoot (Directory Proxy Server 5.*x*)	`/var/opt/mps/serverroot`	`/var/opt/sun/directory-server`

Table 6-1 Directory Proxy Server Directory Paths *(Continued)*

Path Name Variable	Solaris OS	Linux OS
DirServer-base (Directory Proxy Server 6.0)	`/opt/SUNWdsee`	`/opt/sun`

Overview of Directory Proxy Server Upgrades

This section describes the following general aspects of Directory Proxy Server that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5
- Java ES Release 5 Upgrade Roadmap
- Directory Proxy Server Data
- Directory Proxy Server Upgrade Strategy

About Java ES Release 5

Java ES Release 5 Directory Proxy Server represents a major release, being a new product with respect to Release 4 Directory Proxy Server and all previous releases.

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

For more information, see the *Directory Server Enterprise Edition 6 Release Notes*.

Java ES Release 5 Upgrade Roadmap

Table 6-2 shows the supported Directory Proxy Server upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 6-2 Upgrade Paths to Java ES 5 (Release 5): Directory Proxy Server 6.0

Java ES Release	Directory Proxy Server Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Directory Proxy Server 5.2 2005Q4	Direct upgrade: Replace Release 4 with a fresh install and configuration of Release 5.	If backward compatibility desired, manually map previous configuration to new configuration properties.
Release 3	Sun Java System Directory Proxy Server 5.2 2005Q1	Direct upgrade: Replace Release 3 with a fresh install and configuration of Release 5.	If backward compatibility desired, manually map previous configuration to new configuration properties.

Table 6-2 Upgrade Paths to Java ES 5 (Release 5): Directory Proxy Server 6.0 *(Continued)*

Java ES Release	Directory Proxy Server Version	General Approach	Reconfiguration Required
Release 2	Sun Java System Directory Proxy Server 5.2 2004Q2	Direct upgrade: Replace Release 2 with a fresh install and configuration of Release 5.	If backward compatibility desired, manually map previous configuration to new configuration properties.
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.	If backward compatibility desired, manually map previous configuration to new configuration properties.
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.	If backward compatibility desired, manually map previous configuration to new configuration properties.
	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 5.	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 no longer makes use of Directory Server for storing configuration data. Configuration is performed using the new Directory Service Control Center or Directory Server EE command-line utilities.

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

Table 6-3 Directory Proxy Server Data Usage

Type of Data	Location	Usage
Directory Proxy Server configuration data	Directory Proxy Server 5.2: configuration directory Directory Proxy Server 6.0: accessed through Directory Service Control Center and Directory Server EE command-line utilities.	Configuration of Directory Proxy Server

Table 6-3 Directory Proxy Server Data Usage

Type of Data	Location	Usage
Security data	Directory Proxy Server 5.2: SSL configured through Directory Proxy Server Console. Directory Proxy Server 6.0: SSL configured through Directory Service Control Center and Directory Server EE command-line utilities.	Server certificates

Directory Proxy Server Upgrade Strategy

Your strategy for upgrading Directory Proxy Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Directory Proxy Server by presenting issues that might influence your Directory Proxy Server upgrade plan.

Compatibility Issues

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

It is possible, however, to configure Release 5 Directory Proxy Server to be backwardly compatible, that is, to behave like Release 4 Directory Proxy Server and earlier releases. This configuration requires you to manually map previous configuration attributes to Release 5 configuration properties. Details are in the *Directory Server Enterprise Edition 6 Migration Guide,* `http://docs.sun.com/doc/819-0994`.

However, Release 5 Directory Proxy Server has different default behaviors compared to previous versions: it does not allow LDAP controls to pass through the proxy. To reproduce the behavior of previous versions, you can unblock these controls as described in "Post-Upgrade Tasks" on page 129.

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 the following Java ES components:

- **Shared components.** Directory Proxy Server has dependencies on specific Java ES shared components (see Table 1-9 on page 47). Directory Proxy Server upgrades might depend upon upgraded versions of these shared components.
- **Directory Server.** Directory Proxy Server has a co-dependency on Directory Server for providing improved security and performance for LDAP requests. Directory Proxy Server provides front-end access to Directory Server but has no dependency on Directory Server beyond this functional relationship.

Dual Upgrade

Dual upgrades, in which both Directory Proxy Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up existing Directory Proxy Server data.

   See "Directory Proxy Server Data" on page 120 for the location of essential data.

2. Install the new operating system.

   The operating system installation can be on a new system (or a Solaris 10 zone) or it can wipe out the existing file system.

3. Install Release 5 Directory Proxy Server.

4. Create a Release 5 Directory Proxy Server instance and map configuration attributes to the Release 5 Directory Proxy Server properties.

   See the relevant steps in the procedure for "Upgrading Release 4 Directory Proxy Server" on page 126.

In-place Operating System Upgrade

1. Back up existing Directory Proxy Server data.

   See "Directory Proxy Server Data" on page 120 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Directory Proxy Server.

   See the relevant section of this chapter, depending on upgrade path.

Upgrading Directory Proxy Server from Java ES Release 4

This section includes information about upgrading Directory Proxy Server from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Directory Proxy Server Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is achieved by performing a fresh install of Release 5 Directory Proxy Server and then configuring new Directory Proxy Server instances using the Directory Service Control Center or Directory Server EE command-line utilities.
- **Upgrade Dependencies.** Directory Proxy Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Directory Proxy Server.
- **Backward Compatibility.** Release 5 Directory Proxy Server can be configured to be backwardly compatible with its Release 4 version, as explained in "Compatibility Issues" on page 121.
- **Upgrade Rollback.** A rollback of the Release 5 upgrade is achieved by reverting to the previous version, which is left intact by the upgrade to Release 5.
- **Platform Issues.** The general approach for upgrading Directory Proxy Server is the same on both Solaris and Linux operating systems.

Release 4 Directory Proxy Server Upgrade

This section describes how to perform an upgrade of Directory Proxy Server from Java ES Release 4 to Java ES Release 5 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
- Upgrading Release 4 Directory Proxy Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Directory Proxy Server software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Directory Proxy Server Dependencies
- Obtain Required Configuration Information and Passwords

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 6-4 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`
Release 5[1]	`Sun ONE Directory Proxy Server Version 6.0`

[1] The `ldapfwd` command cannot be used to return a version number for Release 5. See "Verifying the Upgrade" on page 128.

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 5. Directory Proxy Server has hard upgrade dependencies on only a few shared components.

When upgrading Directory Proxy Server dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Directory Proxy Server. Upgrade of shared components is normally achieved automatically by the Java ES installer.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Directory Proxy Server are upgraded automatically by the Java ES installer when you perform an upgrade of Directory Proxy Server to Release 5.

2. **Directory Server (soft upgrade dependency)** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, "Directory Server" on page 99. However, Release 5 Directory Proxy Server is supported by Release 4 Directory Server.

Obtain Required Configuration Information and Passwords

Configuration information is preserved through the upgrade process and can be used to map Release 4 configuration attributes to Release 5 configuration properties. See "Compatibility Issues" on page 121.

Upgrading Release 4 Directory Proxy Server

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

Upgrade Considerations

The upgrade of Directory Proxy Server software to Java ES Release 5 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 and re-configured to access the corresponding new Release 5 instance.

- 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 require you to create new Directory Proxy Server instances.

- In Release 4 deployment architectures involving Directory Proxy Server, an Administration Server was used to configure and manage Directory Proxy Server instances. In Release 5 the upgrade of Directory Proxy Server might entail deployment of the Directory Service Control Center, used to configure and manage Directory Proxy Server instances.

Upgrade Procedure

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

1. Log in as root or become superuser.

   ```
   su -
   ```

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

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

3. Perform a fresh install of Release 5 Directory Proxy Server.

   Perform the following steps:

   a. Launch the Java ES installer.

      ```
      cd Java ES Release 5 distribution/os_arch
      ./installer
      ```

      where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

      After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of “upgradable.”)

   b. Select the Directory Proxy Server subcomponent of Directory Server Enterprise Edition.

      You will also need to install the administrative subcomponents (Directory Service Control Panel or command line utilities) you wish to use.

   c. Specify an installation path different from that of any existing Release 4 Directory Proxy Server.

d. Choose to Configure Now or Configure Later.

   It does not matter whether you choose to configure now or to configure later because there is really no configuration required for Directory Proxy Server. However, if you choose to configure now, you are asked if you want to create a new instance. (Once the component is installed, you can create as many Directory Proxy Server instances as you wish.)

e. If needed, select the option to install localized packages.

f. Confirm your installation choices.

   Directory Proxy Server packages will be installed and an installation summary displayed.

g. Exit the Java ES installer.

   If you have not created a new instance, continue to Step 4, otherwise continue with Step 5.

4. Create a Directory Proxy Server instance.

   ```
   DirServer-base/dps6/bin/dpadm create instancePath
   ```

   where *instancePath* is the full path to the Directory Proxy Server instance.

   For information on creating a Directory Proxy Server instance, see the *Directory Server Enterprise Edition 6 Administration Guide*, http://docs.sun.com/doc/819-0995.

5. If desired, map Release 4 configuration attributes to the Release 5 Directory Proxy Server properties.

   For details of the mapping procedure, see the *Directory Server Enterprise Edition 6 Migration Guide*, http://docs.sun.com/doc/819-0994.

Verifying the Upgrade

You can verify successful upgrade of Directory Proxy Server as follows.

1. Start the new Directory Proxy Server instance.

   ```
   DirServer-base/dps6/bin/dpadm start instancePath
   ```

2. Check for the Directory Proxy Server version.

   ```
   DirServer-base/dps6/bin/dpadm --version
   ```

   Output values are shown in Table 6-4 on page 125.

Post-Upgrade Tasks

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

In addition, to reproduce the default behavior of previous versions, LDAP controls must be explicitly allowed to pass through the proxy. You can enabled these controls by setting the `allowed-ldap-controls` property as follows:

```
cd DirServer-base/dps6/bin
```

```
./dpconf set-server-prop
allowed-ldap-controls:auth-request
allowed-ldap-controls:chaining-loop-detection
allowed-ldap-controls:manage-dsa
allowed-ldap-controls:persistent-search
allowed-ldap-controls:proxy-auth-v1
allowed-ldap-controls:proxy-auth-v2
allowed-ldap-controls:real-attributes-only
allowed-ldap-controls:server-side-sorting
```

Rolling Back the Upgrade

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

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 cannot perform a rolling upgrade; the load balancer needs to be shut down and re-configured to access the Release 5 instances. You perform the upgrade of each instance as described in “Release 4 Directory Proxy Server Upgrade” on page 125.

Upgrading Directory Proxy Server from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Directory Proxy Server to Release 5 is the same as that for upgrading Release 4 Directory Proxy Server to Release 5.

To upgrade Release 3 Directory Proxy Server to Release 5, use the instructions in "Upgrading Directory Proxy Server from Java ES Release 4" on page 124, except substitute Release 3 wherever Release 4 is referenced.

Upgrading Directory Proxy Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Directory Proxy Server to Release 5 is the same as that for upgrading Release 4 Directory Proxy Server to Release 5, with the exception that the pre-upgrade tasks should include the synchronizing to Release 5 of all shared components (see Table 1-9 on page 47) and all locally-resident product components upon which Directory Proxy Server depends.

Instructions for synchronizing Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

To upgrade Release 2 Directory Proxy Server to Release 5, use the instructions in "Upgrading Directory Proxy Server from Java ES Release 4" on page 124, except substitute Release 2 wherever Release 4 is referenced.

NOTE If you are upgrading from Release 2 Directory Proxy Server on the Linux platform, then you will have to perform a dual upgrade, in which both Directory Proxy Server *and* the operating system are upgraded (Release 5 Directory Proxy Server is not supported on RHEL 2.1). See "Dual Upgrade" on page 122 for more information.

Chapter 7

Web Server

This chapter describes how to upgrade Web Server to Java ES 5 (Release 5): Sun Java System Web Server 7.0.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Web Server Upgrades" on page 134
- "Upgrading Web Server from Java ES Release 4" on page 138
- "Upgrading Web Server from Java ES Release 3" on page 156
- "Upgrading Web Server from Java ES Release 2" on page 157

NOTE File locations in this chapter are specified with respect to directory paths referred to as *WebServer6-base* (Web Server 6.*x*) and *WebServer7-base and WebServer7Config-base* (Web Server 7.0). At least part of these paths might have been specified as an installation directory when Web Server was initially installed. If not, the Java ES installer assigned a default value.

The default values of these directory paths are shown in the following table.

Table 7-1 Web Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
WebServer6-base	`/opt/SUNWwbsvr`	`/opt/sun/webserver`
WebServer7-base	`/opt/SUNWwbsvr7`	`/opt/sun/webserver7`
WebServer7Config-base	`/var/opt/SUNWwbsvr7`	`/var/opt/sun/webserver7`

Overview of Web Server Upgrades

This section describes the following general aspects of Web Server that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5 Web Server
- Web Server Upgrade Roadmap
- Web Server Data
- Web Server Upgrade Strategy

About Java ES Release 5 Web Server

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

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

For more information on the new administrative infrastructure, see the *Web Server 7.0 Administrator's Guide*, `http://docs.sun.com/doc/819-2629`.

These changes in the Web Server administrative interface have a significant impact on upgrade.

Web Server Upgrade Roadmap

Table 7-2 shows the supported Web Server upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 7-2 Upgrade Paths to Java ES 5 (Release 5): Web Server 7.0

Java ES Release	Web Server Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Web Server 6.1 SP5 2005Q4	Direct upgrade: Fresh install followed by data migration	Migration of instance configuration to new instances.

Table 7-2 Upgrade Paths to Java ES 5 (Release 5): Web Server 7.0

Java ES Release	Web Server Version	General Approach	Reconfiguration Required
Release 3	Sun Java System Web Server 6 2005Q1 Update 1 SP 4	Direct upgrade: Fresh install followed by data migration	Migration of instance configuration to new instances.
Release 2	Sun Java System Web Server 6 2004Q2 Update 1 SP 2 Platform and Enterprise Editions	Direct upgrade: Fresh install followed by data migration	Migration of instance configuration to new instances.
Release 1	Sun ONE Web Server 6.1 (2003Q4)	Direct upgrade not certified: But achieved by performing fresh install followed by data migration.	Migration of instance configuration to new instances.
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 7-3 Web Server Data Usage

Type of Data	Location	Usage
Configuration data	Web Server 6.*x* (Java ES Release 2, 3, and 4): *WebServer6-base*`/https-`*instanceName*`/config/` Web Server 7.0 (Java ES Release 5): *I*nstance Configuration *WebServer7Config-base*`/https-`*configName*`/config`[1]`/` Central Configuration Store Accessed through Web Server Console and through `wadm` command line interface.	Configuration of Web Server instances

[1] Note that the *WebServer7Config-base* path is substantially different from the *WebServer6-base* path.

Web Server Upgrade Strategy

Your strategy for upgrading Web Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Web Server by presenting issues that might influence your Web Server upgrade plan.

Compatibility Issues

Java ES Release 5 Web Server does not introduce any changes in public interfaces and is therefore backwardly compatible with earlier versions in this respect. However, the new administrative interfaces are not backwardly compatible with earlier administrative interfaces. This impacts the upgrade and re-deployment of web applications (including, for example, Java ES components).

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

Table 7-4 Web Server Default Names

Item	Java ES Release 2, 3, and 4 Web Server 6.x Default	Java ES Release 5 Web Server 7.0 Default
Configuration name		*hostName.domainName*
Instance directory path	*WebServer6-base*/ `https-`*hostName.domainName*	*WebServer7Config-base* `https-`*hostName.domainName*
Virtual server name	`https-`*hostName.domainName*	*hostName.domainName*

Web Server Dependencies

Web Server has dependencies on the following Java ES components:

- **Shared components.** Web Server has dependencies on specific Java ES shared components (see Table 1-9 on page 47). Web Server upgrades might depend upon upgraded versions of these shared components.
- **Directory Server.** Web Server has an optional dependency on Directory Server for providing LDAP-based authentication.
- **Web Proxy Server.** Web Server has a co-dependency on Web Proxy Server for providing improved security and performance for HTTP requests.

Dual Upgrade

Dual upgrades, in which both Web Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed using the in-place operating system upgrade approach:

1. Back up existing Web Server data.

   See "Web Server Data" on page 135 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Web Server.

   See the appropriate section of this chapter, depending on upgrade path.

Upgrading Web Server from Java ES Release 4

This section includes information about upgrading Web Server from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Web Server Upgrade

Introduction

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

- **General Upgrade Approach.** The upgrade is performed by doing a fresh install of Release 5 Web Server, migrating Release 4 Web Server instance configuration information to a Release 5 configuration, and then creating Release 5 Web Server instances that correspond to the Release 4 instances.
- **Upgrade Dependencies.** Web Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Web Server. Web Server has hard upgrade dependencies only on NSS and NSPR shared components.
- **Backward Compatibility.** Release 5 Web Server administrative interfaces are not backwardly compatible with the Release 4 version.
- **Upgrade Rollback.** Rollback of the Release 5 upgrade is achieved by reverting to the Release 4 installation, which remains intact.
- **Platform Issues.** The general approach for upgrading Web Server is the same on both Solaris and Linux operating systems.

Release 4 Web Server Upgrade

This section describes how to perform an upgrade of Web Server from Java ES Release 4 to Java ES Release 5 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
- Upgrading Release 4 Web Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Web Server software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Web Server Dependencies
- Back Up Web Server Data
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Web Server by running the Web Server instance server with the `-version` option:

```
WebServer6-base/https-hostName.domainName/start -version
```

Table 7-5 Web Server Version Verification Outputs

Java ES Release	Web Server Version Number
Release 2	6.1SP2
Release 3	6.1SP4
Release 4	6.1SP5
Release 5	7.0

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 5. However, all shared components required by Web Server (see Table 1-9 on page 47) are upgraded automatically by the Java ES installer when you perform an upgrade of Web Server to Release 5.

Back Up Web Server Data

The Web Server upgrade from Release 4 to Release 5 does not modify the existing configuration data; it is left intact. There is no need to back up current data.

Obtain Required Configuration Information and Passwords

You will have to log in as superuser to perform the upgrade and the user account performing migration should have permission to access the existing Web Server installation directories.

Upgrading Release 4 Web Server

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

Upgrade Considerations

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

- **Configure Now or Configure Later.** When performing an upgrade, you specify whether to install Release 5 Web Server using the Configure Now or Configure Later option:
    - Configure Now means the installer will set up an Administration Server or Administration Node, as specified, and also create a default configuration and a corresponding Web Server instance. This approach is useful for installation on a single computer, but the default configuration name might impact the migration of existing instance configurations during upgrade.
    - Configure Later means the installer will perform no configuration: you will have to manually run a `configureServer` script after providing property values to an input file. This approach is useful if you want to automate the installation on multiple computers using scripts that perform silent installs. You also have full control over configuration names and can avoid conflict with the migration of existing instance configurations during upgrade.

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

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

  When performing the migration of a default Release 4 instance configuration to Release 5, there are three approaches you can take, each of which results in a different configuration name.

  The approach you choose can impact the subsequent upgrade of deployed web applications. For example, the upgrade of deployed Java ES components (such as Access Manager and Portal Server) and Sun Java Communications Suite components (such as Communications Express, Instant Messaging, and Delegated Administrator) generally requires that the person performing such upgrades know the name of the Release 5 configuration to which the Release 4 instance configuration has been migrated.

  The three approaches are the following:

  - Specify a *new* configuration name different from the default name, for example "`JavaESapps.`" The `migrate-server` command will then create a new Release 5 configuration named `JavaESapps`.

    Subsequently upgraded web applications would need to be re-deployed to the `JavaESapps` configuration.

  - Do not specify a new configuration name, but delete the default Release 5 instance and configuration (*hostName.domainName*) before running the `migrate-server` command. The `migrate-server` command will then create a new configuration with the default name (*hostName.domainName*). The sequence would be as follows (see `wadm help` for details):

    ```
    wadm delete-instance
    wadm delete-config
    wadm migrate-server
    wadm create-instance
    ```

    Subsequently upgraded web applications would need to be re-deployed to the *hostName.domainName* configuration, which is the same as the default Release 4 instance name.

- Do not specify a new configuration name, and do not delete the default Release 5 configuration. The `migrate-server` command will then create a new configuration with the following name: *hostName.domainName*`-1`.

  Subsequently upgraded web applications would need to be re-deployed to the *hostName.domainName*`-1` configuration.

Whichever approach you take in migrating Release 4 instance configurations, the name of the Release 5 configuration to which the Release 4 instance configuration has been migrated should be communicated to whomever is performing a subsequent upgrade of a web application deployed in that instance.

- **Migration of configuration data.** When migrating Release 4 instance configurations, the following information is automatically migrated:

  - All the configuration information in the Release 4 Web Server instance directory: *WebServer6-base*`/https-`*instanceName*`/config`. This includes configuration information for all web applications deployed in the Release 4 instance (for example, Java ES components such as Access Manager and Portal Server).
  - acl information from *WebServer6-base*`/httpacl`
  - auth-db information from *WebServer6-base*`/userdb`
  - Scheduler information from *WebServer6-base*`/https-admserv/config`
  - Certificate information from *WebServer6-base*`/alias`
  - Search collection information and index files, as specified when you perform the migration.

  The automatic migration does *not* include the following data:

  - docroot content. Instead the new configuration will point to the old docroot and a log message will be recorded in the migration log
  - Webdav data. Webdav collection information will be migrated.
  - 3rd party NSAPI plug-ins will not be migrated. Instead, they will point to the Release 4 file and a log message will be recorded in the migration log
  - Log files
  - Changes to the search collection docroot
  - Command line scripts (`startsvr`, `startsvr.bat`, `stopsvr`, `stopsvr.bat`, `restart`, `reconfig`, `reconfig.bat`).

For details regarding data migration, see the *Web Server 7.0 Installation and Migration Guide*, `http://docs.sun.com/doc/819-2625`.

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

Upgrade Procedure

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. Log in as root or become superuser.

   ```
   su -
   ```

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

   WebServer6-base/https-*instanceName*/stop
   WebServer6-base/https-admserv/stop

3. Perform a fresh install of Release 5 Web Server.

   Perform the following steps:

   a. Launch the Java ES installer.

      `cd` *Java ES Release 5 distribution*/*os_arch*
      `./installer`

      where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

      After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of “upgradable.”)

   b. Select Web Server in the component selection page.

   c. Specify an installation path different from that of Release 4 Web Server.

   d. Choose to Configure Now or Configure Later.

      - If you choose Configure Now, go to Step e.
      - If you choose Configure Later, go to Step f.

e. If you choose to Configure Now, the Java ES installer offers two choices:

- Configure Administration Instance as Administration Server
  Use this choice on the computer that will host the Administration Server that, among other administrative tasks, is required to perform migration of Release 4 instances to Release 5.

- Configure Administration Instance as Administration Node
  Use this choice on a computer that will host a Web Server instance remote from the Administration Server. The administration instance is configured as a node agent that interacts with the Administration Server.

I. Specify the configuration values requested.

You are asked for the host name, HTTP port, admin user name and admin password.

II. Confirm your installation choices.

Web Server packages will be installed and an install summary displayed.

The Java ES installer will create a default configuration named *hostName.domainName* and a corresponding Web Server instance.

III. Exit the Java ES installer and go to Step 4 on page 146.

f. If you choose to Configure Later, the Java ES installer will create a `configureServer` script that you run in Step IV on page 146.

I. Confirm your installation choices.

Web Server packages will be installed and an install summary displayed.

II. Exit the Java ES installer.

III. Set values in the *WebServer7-base*`/setup/WSInstall.properties` file.

Provide values to all the required (non-optional) properties in the following table.

Table 7-6 WSInstall.properties Values

Property	Description
WS_DOCROOT	(Optional) Document location which can host web content files
WS_SERVER_NAME	Host name which can be used to serve HTTP requests
WS_SERVER_USER	Runtime unix user. Valid values can be root, any valid UNIX user, or webservd (default).
WS_HTTP_PORT	Instance port which can be used to listen for HTTP requests
WS_ADMIN_SSL_PORT	Admin SSL port
WS_ADMIN_HOST	Admin host name for admin server tasks
WS_CONFIG_NAME	Config name for this host. This value can be the same value as provided in WS_SERVER_NAME
WS_ADMIN_SERVER_USER	Admin server runtime UNIX user. Valid values: 'root' or the same user as WS_SERVER_USER
WS_ADMIN_LOGIN_USER	Admin server login user name
WS_ADMIN_LOGIN_PASSWORD	Admin server login password
WS_ADMIN_HTTP_PORT	(Optional) Admin Non SSL port. Default: 8800
WS_START_ON_BOOT	(Optional) Start on boot feature (true/false). True will allow server instance and its admin server to auto start after system reboot. Default: false
WS_64BIT_INSTALL	(Optional) Server runtime mode (true/false). True will configure the server in 64 bit mode. (Only for Solaris). False will configure the server in 32 bit mode. (Only for Solaris) Default: false
WS_ADMIN_IS_SERVER_MODE	(Optional) Admin configuration mode. (true/false). True will configure server in admin server mode. False will configure server in admin agent mode. Default: true
WS_REGISTER_ADMIN_AGENT	(Optional) Remote agent registration. (true/false). This is required only if WS_ADMIN_IS_SERVER_MODE is set to false. True will require you to provide the remote admin server host for registration. Default: true
WS_AGENT_SSL_PORT	(Optional) Admin agent SSL port. This is required only if WS_ADMIN_IS_SERVER_MODE is set to false
WS_AGENT_HOST=	(Optional) Admin agent host name. This is required only if WS_ADMIN_IS_SERVER_MODE is set to false

IV. Run the `configureServer` script.

```
WebServer7-base/setup/configureServer
-inputfile WebServer7-base/setup/WSInstall.properties
-logfile WebServer7-base/setup/WSInstall.log
-verbose
```

The `configureServer` script will create a default configuration named *hostName.domainName* and a corresponding Web Server instance.

4. Start the Web Server Administration Server service.

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

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

You can use either the command-line (`wadm`) or graphical user interface administration tools (log in to the Web Server Admin Server GUI). The steps that follow are based on the `wadm` command-line interface.

For example, to migrate an instance named `myinstance` to a new configuration:

```
WebServer7-base/bin/wadm migrate-server --user=admin
--host=localhost --server-root=/opt/SUNWwbsvr
--instance=https-myinstance --config=newconfigname
```

The full command syntax is as follows:

```
WebServer7-base/bin/wadm migrate-server

--user=admin-user [--password-file=admin-pswd-file] [--host=admin-host]
[--echo] [--rcfile=rcfile] [--no-prompt] [--verbose]

[--search-collection-copy-path=searchCollectionPath]
[--log-dir=directory] --serverroot=path
([--all] | [--instance=https-instanceName] [--config=newconfigName])
```

The first set of command options, above, are common to all `wadm` commands, and are documented in Table 7-7, below. The second set of command options are specific to the migrate-server command and are documented in Table 7-8.

Invoking `wadm` with only the first set of command options places you within the `wadm` command shell. Invoking commands within this shell does not require that you specify the common options again.

If you invoke the full wadm commands from outside the shell, you have to specify, at minimum, the --user and --host options. (If you omit the --password-file option, you will be prompted for a password, and if you omit other options, the default value will be assumed.) However, for commands used to illustrate procedures in this chapter, the --user and --host options are not included for the sake of simplicity.

By default, wadm uses the SSL protocol at port 8989.

For full information on wadm commands and options, see the *Web Server 7.0 CLI Reference Manual*, http://docs.sun.com/doc/819-3283.

Table 7-7 wadm Common Command Options

Option	Description
user	Authorized Web Server administrative user ID.
password-file	File containing the password to authenticate the administrative user to the Administration Server. The password file must contain a line WADM_PASSWORD=*password*. If this option is not specified in the command, you will be prompted for the password.
host	Name of the computer where the Administration Server is running. Default: localhost.
echo	Setting this option to true will echo the command line on standard output before executing the command. Default: false
interactive	If this option set to true, the required password options are prompted. Default: true.
rcfile	Startup file to be used to load at the start of wadm. Default: ~/.wadmrc.
no-prompt	If this option set to true, the command will never ask for any user input under any circumstances. For example, the command will simply error out if invoked with missing parameters rather than asking for and waiting for user input.You might want to set to true when using wadm commands with a shell script so that the command always returns rather than wait for user input. Default: false.
verbose	If set to true, verbose listing is displayed. Default: false.

Table 7-8 `wadm migrate-server` Command Options and Operands

Option/Operand	Description
search-collection-copy-path	Specifies the path to which search collection index files will be copied when migrating search collections. The following migration scenarios are possible: If the Web Server 6.*x* search collection path is outside the Web Server 6.*x* instance, then the migrated search collection path will point to the Web Server 6.*x* search collection path, and this option will be disregarded. If the Web Server 6.*x* search collection path is within the Web Server 6.*x* instance, and a valid path is specified for this option, then the search collection index files will be copied to the following directory: *searchCollectionPath*/*configName*/*virtualServerName*/*collectionName*. If the specified path is not valid, an error message will be logged. If the Web Server 6.*x* search collection path is within the Web Server 6.*x* instance but no path is specified for this option, then the search collection index files will not be copied. A message will be written to the migration log asking the user to manually copy the search collection index files using the `wadm add-documents` command. In this case, the migrated search collection path will be the following: *WebServer7Config-base*/`https-`*configName*`/config/collections`/ *virtualServerName*/*collectionName*.
log-dir	The location of the migration log. Default: *WebServer7Config-base*/`admin-serv/logs`
serverroot	Installation location (directory) where Web Server 6.*x* version is installed: same as *WebServer6-base*.
all	If set to `true`, all Web Server 6.*x* instance configurations are migrated to Web Server 7.0 configurations of the same names as the instances. If a configuration of that name already exists, *instanceName*`-1` is used as the configuration name. Default: `false`.
instance	If instance configurations are to be individually migrated (all=`false`), Name of the Web Server 6.*x* instance configuration to be migrated (in the form:`https-`*instanceName*). The default Web Server 6.x instance name is *hostName.domainName*
config	Name of the configuration to which the specified Web Server 6.*x* instance configuration is to be migrated. The default is to use the *instanceName* of the Web Server 6.*x* instance configuration. However, if a configuration of that name already exists, the command will append an integer to the name. This is a likely scenario if the default Web Server 6.*x* instance configuration is being migrated.

In using the `migrate-server` command, please keep in mind the following considerations:

- If you want to migrate multiple Release 4 instance configurations, you can either run the `migrate-server` command multiple times with different `--instance` values and corresponding `--config` arguments, or use the `--all` option to migrate them all at once.
- For every invocation of the migrate-server command, the migration will create a log file of the following name in a directory specified by the `--log-dir` option (or in the default *WebServer7Config-base*`/admin-server/logs` directory):

  `MIGRATION_`*yyyymmddhhmmss*`.log`

  If you select the `--all` option, then the log file will store migration information for all migrated instances.
- For data that is not migrated by the `migrate-server` command (see "Upgrade Considerations" on page 140), you have to perform the migration manually (see "Post-Upgrade Tasks" on page 151).

6. Create Release 5 Web Server instances.

You must create a new Release 5 instance for each Release 4 instance configuration migrated in Step 5.

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

b. Run the `create-instance` command.

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

Common command options are documented in Table 7-7 on page 147. Options specific to the `create-instance` command are documented in the following table.

Table 7-9`wadm create-instance` Command Options and Operands

Option/Operand	Description
config	The name of the Release 5 configuration that the instance should point to.
nodehost	Name of the computer on which the instance is being created. You can specify multiple computers as a space-separated list of *hostName.domainName*, thereby creating multiple identical instances.

The `create-instance` command creates an instance directory at
WebServer7Config-base`/https-`*configName*
on the specified nodes and deploys the configuration to the corresponding instance directories.

7. Start each Release 5 instance.

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

   The `startserv` script is created when the instance is created. If the instance starts without any problem, then you see a message saying "successful server startup." The default URL for the instance will be displayed.

Verifying the Upgrade

You can verify the upgrade of Web Server to Release 5 by performing the following steps:

1. Check the newly created migration log file for any `ERROR` messages.

   If needed, make manual changes (see "Post-Upgrade Tasks" on page 151).

2. Verify the Release 5 Web Server instances.

   From a web browser access the following URL and make sure you get the welcome page:

   http://*hostName.domainName*:*port*

   where the fully-qualified host name and port correspond to each instance.

3. Run the Web Server instance with the `-version` option:

   WebServer7Config-base`/https-`*configName*`/bin/startserv -version`

   See Table 7-5 on page 139 for version output values.

Post-Upgrade Tasks

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

Please note the post-upgrade procedures to address the following situations:

- Migrating Web Server 6.1 docroot content
- Migrating webdav collection information
- Migrating Log files
- Migrating 3rd party NSAPI plug-ins
- Changing the search collection document root
- Customizing command-line scripts

Migrating Web Server 6.1 docroot content

1. Copy the Web Server 6.1 docroot content to wherever you want.
2. Update the new document root path using the following command:

   ```
   WebServer7-base/bin/wadm set-virtual-server-prop
   --config=configName --vs=virtualServerName
   document-root=new docroot path
   ```

   Common command options are documented in Table 7-7 on page 147. Options specific to the `set-virtual-server-prop` command are documented in the following table.

Table 7-10 `wadm set-virtual-server-prop` Command Options and Operands

Option/Operand	Description
config	Name of the Release 5 configuration for which the new document root path is being set.
vs	The name of the virtual server to which the migrated document root corresponds.
document-root	The path to the new document root directory.

3. Redeploy the configuration to the relevant Web Server instances.

   ```
   WebServer7-base/bin/wadm deploy-config
   [--force] [--restart] [--no-reconfig]
   configName
   ```

   Common command options are documented in Table 7-7 on page 147. Options specific to the `deploy-config` command are documented in the following table.

Table 7-11 `wadm deploy-cofig` Command Options and Operands

Option/Operand	Description
force	If `true`, forces the overwriting of an instance configuration that has been manually modified since the previous configuration deployment. Default: `false`
restart	If `true`, running instances will be restarted to pick up configuration settings in the deployed configuration. Default: `false`
no-reconfig	if `true`, running instances will not pick up configuration settings in the deployed configuration until the instance is restarted. Default: `false`
configName	Name of the Release 5 configuration that is being deployed to a Web Server instance whose instance name corresponds to the configuration name.

Migrating webdav collection information

No extra manual migration needed. Just updating the docroot path is enough.

Migrating Log files

Copy these files to a known location if you want to save them (otherwise they will be deleted should you remove the Release 4 installation).

Migrating 3rd party NSAPI plug-ins

1. Copy the library files from their Release 4 location to the *WebServer7-base*`/lib` directory.

2. Export the `magnus.conf` and `obj.conf` configuration files to a temporary directory.

   ```
   WebServer7-base/bin/wadm get-config-file --config=configName
   magnus.conf > /tmp/magnus.conf
   ```

```
WebServer7-base/bin/wadm get-config-file --config=configName
obj.conf > /tmp/obj.conf
```

Common command options are documented in Table 7-7 on page 147.

3. Modify `magnus.conf` and obj.conf files as specified in 3rd party NSAPI plugin documentation.

4. Import the `magnus.conf` and `obj.conf` configuration files from the temporary directory.

```
WebServer7-base/bin/wadm set-config-file --config=configName
--upload-file=/tmp/magnus.conf magnus.conf
```

```
WebServer7-base/bin/wadm set-config-file --config=configName
--upload-file=/tmp/obj.conf obj.conf
```

Common command options are documented in Table 7-7 on page 147.

5. Redeploy the modified configuration to the relevant Web Server instances.

```
WebServer7-base/bin/wadm deploy-config
[--force] [--restart] [--no-reconfig]
configName
```

Command options are documented in Table 7-11 on page 152.

Changing the search collection document root

The migrate-server command has an option for migrating search collection information, however you might want to change the search collection document root, as follows:

1. If the document root for the search collection is different from that used for Release 4, use the following command to set the document root for the search collection:

```
WebServer7-base/bin/wadm set-search-collection-prop
--config=configName --vs=virtualServerName
--collection-name=searchCollectionName document-root=new docroot path for
the search collection
```

Common command options are documented in Table 7-7 on page 147. Options specific to the `set-search-collection-prop` command are documented in the following table.

Table 7-12 `wadm set-search-collection-prop` Command Options and Operands

Option/Operand	Description
config	Name of the Release 5 configuration for which the document root of the search collection is being set.
vs	The name of the virtual server to which the search collection corresponds.
collection-name	The name of the search collection for which a new document root path is being set.
document-root	The path to the new document root directory for the search collection.

2. Redeploy the configuration to the relevant Web Server instances.

   ```
   WebServer7-base/bin/wadm deploy-config
   [--force] [--restart] [--no-reconfig]
   configName
   ```

   Command options are documented in Table 7-11 on page 152.

Customizing command-line scripts

If scripts such as `startsvr`, `startsvr.bat`, `stopsvr`, `stopsvr.bat`, `restart`, `reconfig`, and `reconfig.bat` have been customized, then you will have to perform the same customizations on the Release 5 default scripts, located in the following directory: *WebServer7-base*`/bin`.

Rolling Back the Upgrade

Release 4 Web Server was left intact by the fresh installation of Release 5 and subsequent migration of Web Server instance configurations. Hence, the rollback of Release 5 Web Server consists of the following steps for reverting back to Release 4.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop all running Web Server instance one by one.

   ```
   WebServer7Config-base/https-configName/bin/stopserv
   ```

   If the server was stopped properly then you will see a message “server has been shutdown”.

3. Remove the Release 5 Web Server installation.

   You have to remove all Release 5 instances and migrated configurations:

   a. Delete all Release 5 instances.

      ```
      WebServer7-base/bin/wadm delete-instance --user ...
          --config=configName hostName.domainName
      ```

   b. Delete all Release 5 configurations.

      ```
      WebServer7-base/bin/wadm delete-config --user ... configName
      ```

4. Restart the Web Server instances that were stopped when upgrading Web Server, as described in "Upgrade Procedure" on page 143.

Upgrading Web Server from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Web Server to Release 5 is the same as that for upgrading Release 4 Web Server to Release 5.

To upgrade Release 3 Web Server to Release 5, use the instructions in "Upgrading Web Server from Java ES Release 4" on page 138, except substitute Release 3 wherever Release 4 is referenced.

Upgrading Web Server from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Web Server to Release 5 is the same as that for upgrading Release 4 Web Server to Release 5.

To upgrade Release 2 Web Server to Release 5, use the instructions in "Upgrading Web Server from Java ES Release 4" on page 138, except substitute Release 2 wherever Release 4 is referenced.

NOTE If you are upgrading from Release 2 Web Server on the Linux platform, then you will have to perform a dual upgrade, in which both Web Server *and* the operating system are upgraded (Release 5 Web Server is not supported on RHEL 2.1). See "Dual Upgrade" on page 137 for more information.

Chapter 8

Java DB

This chapter describes how to upgrade Java DB to Java ES 5 (Release 5): Java DB 10.1.3.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. 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 *JavaDB-base*. This path was set by the Java ES installer when Java DB was installed.

The values of this directory path is shown in the following table.

Table 8-1 Java DB Directory Paths

Path Name Variable	Solaris OS	Linux OS
JavaDB-base	`/opt/SUNWjavadb`	`/opt/sun/javadb`

Overview of Java DB Upgrades

This section describes the following general aspects of Java DB that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5
- Java ES Release 5 Upgrade Roadmap
- Java DB Data
- Java DB Upgrade Strategy

About Java ES Release 5

Java ES Release 5 Java DB is the first release to be delivered as a Java ES product component; Java DB was first released as a shared component named DerbyDatabase, included in Java ES Release 4.

Release 5 Java DB represents a minor release with respect to the Release 4 version. It includes some improved functionality, updated interfaces, and selected bug fixes.

Java ES Release 5 Upgrade Roadmap

Table 8-2 shows the supported Java DB upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 8-2 Upgrade Paths to Java ES 5 (Release 5): Java DB 10.1.3

Java ES Release	Java DB Version	General Approach	Reconfiguration Required
Release 4	Derby Database 10.0.2	Direct upgrade: Replace Release 4 with a fresh install. Persistent data is not affected.	None

Java DB Data

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

Table 8-3 Java DB Data Usage

Type of Data	Location	Usage
Configuration data	Instance configuration is application-specific and is stored in the Java DB database.	Configuration of Java DB instance
Persistent data	Database directories and their contents are application-specific. Their location is specified by the database connection URL, `jdbc:derby:`*full path to database*.	Database and user certificates

Java DB Upgrade Strategy

Your strategy for upgrading Java DB generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Java DB by presenting issues that might influence your Java DB upgrade plan.

Compatibility Issues

Release 5 Java DB is backwardly compatible with the Release 4 version.

Dependencies

Java DB has a dependency only on the J2SE shared component (see Table 1-9 on page 47).

Dual Upgrade

Dual upgrades, in which both Java DB and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up existing Java DB data.

   See "Java DB Data" on page 161 for the location of essential data.

2. Install the new operating system.

   The operating system installation can be on a new system (or a Solaris 10 zone) or it can wipe out the existing file system.

3. Install Release 5 Java DB.

4. Restore the Java DB data that was backed up in Step 1.

In-place Operating System Upgrade

1. Back up existing Java DB data.

   See "Java DB Data" on page 161 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Java DB.

   See the "Upgrading Java DB from Java ES Release 4" on page 163.

   Java DB data should remain unaffected by the upgrade.

Upgrading Java DB from Java ES Release 4

This section includes information about upgrading Java DB from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Java DB Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is achieved by performing a fresh install of Release 5 Java DB, replacing the Release 4 version. Release 4 data and configuration remain intact.
- **Upgrade Dependencies.** Java DB has a hard upgrade dependency on the J2SE shared component (see Table 1-9 on page 47), which is automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Java DB.
- **Backward Compatibility.** Release 5 Java DB is fully compatible with Release 4.
- **Upgrade Rollback.** A rollback of the Release 5 upgrade cannot be achieved except by reverting to a backed up Release 4 installation.
- **Platform Issues.** The general approach for upgrading Java DB is the same on both Solaris and Linux operating systems.

Release 4 Java DB Upgrade

This section describes how to perform an upgrade of Java DB from Java ES Release 4 to Java ES Release 5 on both the Solaris and Linux platform. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading Release 4 Java DB
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Java DB software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Java DB Dependencies
- Back Up Java DB Data
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Java DB using the following command:

```
java -cp JavaDB-base/lib/derby.jar org.apache.derby.tools.sysinfo
```

The version information (for the `derby.jar` file) is shown in the following table:

Table 8-4 Java DB Version Verification Outputs

Java ES Release	Java DB Version Number
Release 4	`10.0.2.1`
Release 5	`10.1.3.1`

Upgrade Java DB 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 5. However, all shared components required by Java DB are upgraded automatically by the Java ES installer when you perform an upgrade of Java DB to Release 5.

Back Up Java DB Data

The Java DB upgrade from Release 4 to Release 5 does not modify configuration data or persistent data. However, for the sake of security, you should back up your entire Java DB installation and your data. See Table 8-3 on page 161.

Obtain Required Configuration Information and Passwords

No configuration information or password is required to upgrade Java DB.

Upgrading Release 4 Java DB

This section describes the upgrade procedure on Solaris and Linux platforms.

Upgrade Procedure

The procedure documented below applies to all Java DB instances residing locally on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Release 4 Java DB.

   If you have a network server running, use the following command:

   ```
   java -cp JavaDB-base/lib/derby.jar:JavaDB-base/lib/derbynet.jar
   org.apache.derby.drda.NetworkServerControl shutdown
   ```

   Otherwise, simply shut down all applications using Java DB.

3. Perform a fresh install of Release 5 Java DB.

   Perform the following steps:

a. Launch the Java ES installer on the computer hosting Release 4 Java DB.

```
cd Java ES Release 5 distribution/os_arch
./installer
```

where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

b. Select Java DB from the component selection page.

The previous installation will be overwritten.

c. Select the Configure Later option.

Configure Now is not supported.

d. If needed, select the option to install localized packages.

e. Exit the Java ES installer when installation is complete.

4. Start Release 5 Java DB.

If you are using a network server, use the following command:

```
java -jar <JavaDB-base>/lib/derbynet.jar start
```

Otherwise, simply start any application using Java DB in embedded mode.

Verifying the Upgrade

You can verify successful upgrade of Java DB using the following command:

```
java -cp JavaDB-base/lib/derby.jar org.apache.derby.tools.sysinfo
```

See Table 8-4 on page 164 for output values (for the version of the `derby.jar` file).

Post-Upgrade Tasks

When upgrading Java DB from Release 4 to Release 5, you must convert data from the Java DB 10.0 disk format to the 10.1 format. To perform this conversion, connect to the database with `upgrade=true` appended to the JDBC URL. For example:

```
java -cp JavaDB-base/lib/derbytools.java:JavaDB-base/lib/derby.jar
org.apache.derby.tools.ij

ij version 10.1

ij> connect 'jdbc:derby:/databasePath;upgrade=true';

ij> exit;
```

Rolling Back the Upgrade

A rollback of the Release 5 upgrade cannot be achieved except by reverting to a backup Release 4 installation and its data.

Multiple Instance Upgrades

In some deployment architectures Java DB is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Java DB instances running on multiple computers with a load balancer to distribute the load.

You perform the upgrade of Java DB on each computer as described in "Release 4 Java DB Upgrade" on page 164.

Chapter 9

High Availability Session Store

This chapter describes how to upgrade High Availability Session Store to Java ES 5 (Release 5): High Availability Session Store (HADB) 4.4.3.

The chapter provides a general overview of upgrade issues before covering the different upgrade paths supported by Release 5. 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 installer assigned a default value.

The default value of *HADB-base* does not depend on operating system platform, as shown in the following table.

Table 9-1 HADB Directory Paths

Path Name Variable	Solaris OS	Linux OS
HADB-base Java ES installer	`/opt/SUNWhadb/`*version_number*	`/opt/SUNWhadb/`*version_number*
HADB-base standalone Application Server 8.2 EE installer	`/opt/SUNWappserver/appserver /hadb/`*version_number*	`/opt/SUNWappserver/appserver /hadb/`*version_number*

Overview of HADB Upgrades

This section describes the following general aspects of HADB that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5 HADB
- HADB Upgrade Roadmap
- HADB Data
- HADB Upgrade Strategy

About Java ES Release 5 HADB

Java ES Release 5 versions of HADB represents minor user enhancements with respect to Release 4 HADB.

HADB Upgrade Roadmap

Table 9-2 shows the supported HADB upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 9-2 Upgrade Paths to Java ES 5 (Release 5): HADB 4.4.3

Java ES Release	HADB Version	General Approach	Reconfiguration Required
Release 4	HADB 4.4.2 (2005Q4)	Direct upgrade: An online and an offline upgrade approach are both available.	None
Release 3	HADB 4.4.1 (2005Q1)	Direct upgrade: An online and an offline upgrade approach 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 9-3 HADB Data Usage

Type of Data	Location	Usage
Dynamic application data	`/var/opt/SUNWhadb`	High availability session store
Configuration data	`/etc/opt/SUNWhadb` `/etc/init.d/ma-initd`	High availability server configuration

HADB Upgrade Strategy

Your strategy for upgrading HADB generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to HADB by presenting issues that might influence your HADB upgrade plan.

Compatibility Issues

Release 5 HADB is backwardly compatible with HADB provided with Java ES Release 4.

HADB Dependencies

Release 5 HADB has dependencies only on the J2SE shared component: Java™ 2 Platform, Standard Edition (J2SE™) Version 1.4 or later.

Dual Upgrade

Dual upgrades, in which both HADB software and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) are performed in the context of Application Server dual upgrades. See "Dual Upgrade" on page 212 for information about Application Server dual upgrades.

Upgrading HADB from Java ES Release 4

This section includes information about upgrading HADB from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 HADB Upgrade

Introduction

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

- **General Upgrade Approach.** Upgrades consist of removing the Java ES Release 4 HADB packages and adding the Java ES Release 5 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 has no hard upgrade dependencies. HADB requires J2SE Version 1.4 or later, meaning that it has a soft upgrade dependency on J2SE.
- **Backward Compatibility.** HADB provided with Java ES Release 5 is backwardly compatible with HADB provided with Java ES Release 4.
- **Upgrade Rollback.** Rollback from the Java ES Release 5 upgrade to Java ES Release 4 is achieved by restoring Release 4 version packages.
- **Platform Issues.** The general approach for upgrading HADB is the same on both Solaris and Linux operating systems.

Release 4 HADB Upgrade

This section describes how to perform an upgrade of HADB from Java ES Release 4 to Java ES Release 5 on both the Solaris and Linux platform. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading Release 4 HADB
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade HADB software you should perform the following tasks:

- Verify Current Version Information
- Upgrade HADB Dependencies
- Back Up Directory Data and Configuration Files
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of HADB using standard version checking utilities. For example:

On Solaris:

```
pkgparam -v SUNWhadba
```

On Linux:

```
rpm -qi sun-hadb-a-4.4.3-5.i386.rpm
```

Table 9-4 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

Release 5	VERSION=4.4.3,REV=5
	SUNW_PRODVERS=4.4.3

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 5. However, all shared components required by HADB (namely J2SE) are upgraded automatically by the Java ES installer when you perform an upgrade of HADB to Release 5.

Back Up Directory Data and Configuration Files

The HADB upgrade from Java ES Release 4 to Java ES Release 5 does not in itself modify HADB dynamic data. However, you can back up the Java ES Release 4 packages in case you need to roll back the upgrade.

Also, back up the following files if you have made any modifications to them since the previous HADB installation.

```
/etc/opt/SUNWhadb/mgt.cfg
/etc/init.d/ma-initd
```

Obtain Required Configuration Information and Passwords

HADB upgrade requires you to know the superuser password.

Upgrading Release 4 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 5 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 5 upgrade packages for Solaris and Linux platforms are shown in the following table. Solaris packages are listed in their installation sequence.

Table 9-5 Package Versions for Upgrading HADB on Solaris Platforms

Solaris Packages	Linux Packages
SUNWhadba	sun-hadb-a--4.4.3-5.i386.rpm
SUNWhadbc	sun-hadb-c-4.4.3-5.i386.rpm
SUNWhadbe	sun-hadb-e-4.4.3-5.i386.rpm
SUNWhadbi	sun-hadb-i-4.4.3-5.i386.rpm
SUNWhadbj	sun-hadb-j-4.4.3-5.i386.rpm
SUNWhadbm	sun-hadb-m-4.4.3-5.i386.rpm
SUNWhadbs	sun-hadb-s-4.4.3-5.i386.rpm
SUNWhadbv	sun-hadb-v-4.4.3-5.i386.rpm
SUNWhadbx	sun-hadb-x-4.4.3-5.i386.rpm

Online Upgrades of HADB

When you perform an online upgrade of HADB, you first install, start up, and verify Release 5 HADB on each server in the cluster being upgraded. Each server then un-registers from the earlier installation of HADB and 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 8.2 High Availability Administration Guide*, `http://docs.sun.com/doc/819-4740/6n4r9qo7n?a=view`

Offline Upgrades of HADB

An offline upgrade of HADB is available when upgrading from either Java ES Release 3 or Release 4.

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 5 (Release 5) distribution, shown in Table 9-5 on page 175.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Shut down all HADB services.

   a. List all databases that are running.

      ```
      HADB-base/bin/hadbm   list
      ```

   b. Shut down each of the listed databases.

      ```
      HADB-base/bin/hadbm  stop  databaseName
      ```

      Ignore the message if a database is already in the stopped state.

   c. Shut down the HADB management agent on every host running a management agent:

      ```
      /etc/init.d/ma-initd   stop
      ```

      (The `ma-initd` script is located in *HADB-base*`/bin` if you have installed HADB using the standalone Application Server installer instead of the Java ES installer.)

3. Launch the Java ES installer.

   ```
   cd Java ES Release 5 distribution/os_arch
   ./installer
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

   After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of “upgradable.”)

4. Select High Availability Session Store 4.4 in the component selection page.

   If you have already selected Application Server Enterprise Edition 8.2, then HADB is automatically selected.

5. Confirm your upgrade choice.

   HADB packages will be upgraded and an upgrade summary displayed.

6. Exit the Java ES installer.

7. Restore the files backed up in "Back Up Directory Data and Configuration Files" on page 174.

8. Verify that the symbolic link /opt/SUNWhadb/4 now points to *HADB-base*.

   For example, for the default *HADB-base*:

   ```
   ls   -l   /opt/SUNWhadb/4
   lrwxrwxrwx  1 root   other  7 Jul 7 23:18 /opt/SUNWhadb/4 ->
       4.4.3-5/
   ```

9. Restart the HADB management agents that were shut down in Step 2:

   ```
   /etc/init.d/ma-initd   start
   ```

   (The ma-initd script is located in *HADB-base*/bin if you have installed HADB using the standalone Application Server installer instead of the Java ES installer.)

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,

   ```
   HADB-base/bin/ma -V
   Sun Java System High Availability Database 4.4 Database Management
   Agent
   Version     : 4.4.3.5 [V4-5-3-5 2006-03-31 13:59:50 pakker@astra07]
   (SunOS_5.9_sparc)
   ```

2. Check whether the database is running.

   ```
   HADB-base/bin/hadbm status -n databaseName
   ```

   For example, for a database named Example DB, enter the following commands.

   ```
   HADB-basebin/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.

   ```
   HADB-base/bin/hadbm get PackageName databaseName
   ```

   For example, for a database named Example DB, enter the following commands.

   ```
   HADB-base/bin/hadbm get PackageName ExampleDB
   Attribute   Value
   PackageName V4.4.3.5
   ```

   The above command displays the current version of HADB. For a detailed listing, issue the following command:

   ```
   HADB-base/bin/hadbm get --all ExampleDB
   ```

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrading HADB from Java ES Release 4” on page 172.

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 had previously backed up, as described in “Back Up Directory Data and Configuration Files” on page 174.

Upgrading HADB from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) HADB to Release 5 is the same as that for upgrading Release 4 HADB to Release 5.

To upgrade Release 3 HADB to Release 5, use the instructions in "Upgrading HADB from Java ES Release 4" on page 172, except substitute Release 3 wherever Release 4 is referenced.

Chapter 10

Message Queue

This chapter describes how to upgrade Message Queue software from previous Java ES versions to Java ES 5 (Release 5): Sun Java System Message Queue 3.7 UR1.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Message Queue Upgrades" on page 182
- "Upgrading Message Queue from Java ES Release 4" on page 187
- "Upgrading Message Queue from Java ES Release 3" on page 193
- "Upgrading Message Queue from Java ES Release 2" on page 194

NOTE File locations in this chapter are specified with respect to a fixed directory path referred to as *MessageQueue-base*.

The value of *MessageQueue-base* depends on operating system platform:, as shown in the following table.

Table 10-1 Message Queue Directory Paths

Path Name Variable	Solaris OS	Linux OS
MessageQueue-base	`/usr/bin`	`/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 5 (Release 5):

- About Java ES Release 5 Message Queue
- Message Queue Upgrade Roadmap
- Message Queue Data
- Message Queue Upgrade Strategy

About Java ES Release 5 Message Queue

Java ES Release 5 Message Queue represents minor upgrade with respect to Release 4. It includes mostly code fixes with no minor features enhancements.

Message Queue software has historically included two editions, a Platform Edition and an Enterprise Edition, each corresponding to a different feature set and licensed capacity. Enterprise Edition was for deploying and running messaging applications in an enterprise production environment. Platform Edition was mainly for developing, debugging, and load testing messaging applications and components. With Release 5 Message Queue, the Platform Edition is deprecated and Message Queue includes all Enterprise Edition features. An upgrade from an earlier Java ES release version to Release 5 converts any installed Platform Edition to full Message Queue enterprise-level features.

Message Queue Upgrade Roadmap

Table 10-2 shows the supported Message Queue upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 10-2 Upgrade Paths to Java ES 5 (Release 5): Message Queue 3.7 UR1

Java ES Release	Message Queue Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Message Queue 2005Q4 (3.6 SP3)) Enterprise Edition only	Direct upgrade: Performed using Java ES installer.	Data conversion performed automatically.
Release 3	Sun Java System Message Queue 2005Q1 (3.6) Enterprise Edition only	Direct upgrade: Performed using Java ES installer.	Data conversion performed automatically.
Release 2	Sun Java System Message Queue 2004Q2 (3.5 SP1) 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 2003Q4 (3.0.1 SP2) Platform and Enterprise Editions	Direct upgrade not certified: But can be performed using the `mqupgrade` script.[1]	Performed automatically on Solaris platforms, and an `mqmigrate` script is available on Linux platforms.
Pre-dates Java ES releases	Sun Java System Message Queue 3.0.*x* and earlier versions Platform and Enterprise Editions	Direct upgrade not certified: But can be performed using Java ES installer.	

[1] Back up and then restoration of the following files might be required before and after running the `mqupgrade` script: for example (on Solaris OS): restoring `/etc/imq/passwd` and `/etc/imq/accesscontrol.properties` to `/var/imq/instances/`*instanceName*`/etc/`

In addition to the Java ES releases of Message Queue shown in Table 10-2, Message Queue is also bundled with Solaris OS software. Upgrade of the bundled versions of Message Queue to Release 5 can be performed using the Java ES installer.

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 10-3 shows the location of data on Solaris systems. The location on Linux systems is similar, but depends on the version of Message Queue:

- Release 2: replace `/imq` in the table by `/opt/imq`
- Release 3 and later: replace `/imq` in the table by `/opt/sun/mq`

For more information, see the *Message Queue 3.7 UR1 Administration Guide,* `http://docs.sun.com/doc/819-4467/6n6k98brl?a=view.`

In Table 10-3, *instanceName* identifies the name of the Message Queue broker instance with which the data is associated.

Table 10-3 Message Queue Data Usage (Solaris OS)

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	Release 2, Release 3, & Release 4:
	`/var/imq/instances/`*instanceName*`/fs350/`
	Release 5:
	`/var/imq/instances/`*instanceName*`/fs370/`
	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.

Message Queue Upgrade Strategy

Your strategy for upgrading Message Queue generally depends on the many issues discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Message Queue by presenting issues that might influence your Message Queue upgrade plan.

Compatibility Issues

Release 5 Message Queue introduces no new incompatibilities over Release 3 or Release 4. However, there are significant compatibility issues with respect to Release 2 and earlier versions. These are discussed in "Release 2 Compatibility Issues" on page 195.

In addition, as a general rule, if you mix Release 4 and earlier Message Queue brokers and Release 5 Message Queue brokers in a cluster, the master broker must be the earlier release broker, and the cluster will run as the earlier release Message Queue cluster.

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 1-9 on page 47).

- **Directory Server.** Message Queue has an optional dependency on Directory Server: you can configure Message Queue to store administered objects and/or user data in an LDAP directory (Directory Server) rather than locally.

- **Web Container.** Message Queue has an optional dependency on Web Server, Application Server, or a third-party web container to support HTTP messaging between client and broker.

- **Databases.** Message Queue has an optional dependency on Java DB (or third-party databases) to provide JDBC-accessible data store, rather than a flat-file message store, for the Message Queue persistence layer.

- **Sun Cluster**. Message Queue has an optional dependency on Sun Cluster to provide high availability support.

Dual Upgrade

Dual upgrades, in which both Message Queue and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up existing Message Queue data.

   See "Message Queue Data" on page 184 for the location of essential data.

2. Install the new operating system.

   The operating system installation can be on a new system (or a Solaris 10 zone) or it can wipe out the existing file system.

3. Install Release 5 Message Queue.

4. Restore or migrate the Message Queue data that was backed up in Step 1.

   When upgrading from Release 2 Message Queue on Linux, the data is restored to the Release 5 location.

In-place Operating System Upgrade

1. Back up existing Message Queue data.

   See "Message Queue Data" on page 184 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Message Queue.

   See the appropriate section of this chapter, depending on your upgrade path. the upgrade should leave the existing Message Queue data in tact.

   When upgrading from Release 2 Message Queue on Linux, however, the data must be moved to the Release 5 location.

Upgrading Message Queue from Java ES Release 4

This section includes information about upgrading Message Queue from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Message Queue Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is performed using the Java ES installer. The installer migrates configuration data from Release 4 automatically. In addition, any dynamic application data associated with Release 4 will be converted automatically the first time `imqbrokerd` is run. For a file-based store, this means the contents of the `fs350` directory will be copied to a new `fs370` directory. For a JDBC store, a simple version update will occur to the existing database tables.
- **Upgrade Dependencies.** Message Queue has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Message Queue.

  In addition, Release 5 Message Queue has dependencies on Java ES product components, as described in "Message Queue Dependencies" on page 185. However, upgrade of these components is not required to upgrade Message Queue to Release 5.
- **Backward Compatibility.** Release 5 Message Queue is fully compatible with Release 4, with respect to protocols, broker compatibility, administered objects, administration tools, and client applications.
- **Upgrade Rollback.** There is no utility for rolling back the Message Queue upgrade to Release 4. You have to remove the upgraded components and manually restore the previous version and configuration data.
- **Platform Issues.** The approach for upgrading Message Queue is the same on both Solaris and Linux operating systems.

Release 4 Message Queue Upgrade

This section describes how to perform a Message Queue upgrade from Java ES Release 4 to Java ES Release 5:

- Pre-Upgrade Tasks
- Upgrading Release 4 Message Queue
- Verifying the Message Queue Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Message Queue software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Message Queue Dependencies
- Back Up Message Queue

Verify Current Version Information

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 10-4 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`

Release 5	`Sun Java(tm) System Message Queue 3.7`
	`Version: 3.7 UR1`

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 5. Message Queue has hard upgrade dependencies on only a couple of shared components.

When upgrading Message Queue dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Message Queue. Upgrade of shared components is achieved automatically by the Java ES installer.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Message Queue are upgraded automatically by the Java ES installer when you perform an upgrade of Message Queue to Release 5.

2. **Sun Cluster (soft upgrade dependency).** Instructions for upgrading Sun Cluster to Release 5 are provided in Chapter 3, "Sun Cluster Software" on page 75.

3. **Directory Server (soft upgrade dependency).** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, "Directory Server" on page 99.

4. **Java DB (soft upgrade dependency).** You need to perform a fresh install of Release 5 Java DB when upgrading Message Queue.

5. **Web Container Software (soft upgrade dependency).** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, "Web Server" on page 133 and Chapter 11, "Application Server" on page 205, respectively.

Back Up Message Queue

It is always a good practice to back up application data in a production environment before performing an upgrade. Note the location of the persistent store for dynamic application data indicated in Table 10-3 on page 184.

Upgrading Release 4 Message Queue

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 5 Message Queue.

4. Log in as Root.

   ```
   su -
   ```

5. Launch the Java ES installer.

   ```
   cd Java ES Release 5 distribution/os_arch
   ./installer
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

   After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

6. Select Message Queue in the component selection page.

7. Confirm your upgrade choice.

   Message Queue packages will be upgraded and an upgrade summary displayed.

8. Exit the Java ES installer.

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. There has been no change to the HTTP tunneling servlet between Release 4 and Release 5. For more information on HTTP support, see the *Message Queue 3.7 UR1 Administration Guide*, `http://docs.sun.com/doc/819-4467`.

Also, if you are sure you will not need to roll back the upgrade, you can remove the Release 4 file-based data store located in the `fs350` directory (see Table 10-3 on page 184).

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:

   On Solaris:

   ```
   pkginfo | grep -i "message queue"
   ```

   On Linux:

   ```
   rpm -qa | grep mq
   ```

6. Remove the Message Queue packages, using the following command:

   On Solaris:
   `pkgrm` *packageName*
   where *packageName* is any of the Message Queue packages. To remove multiple packages, separate the package names by a space.

   On 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 4 Message Queue.

   Use the Java ES Release 4 installer.

10. Restore Release 4 Message Queue data backed up in "Back Up Message Queue" on page 189.

    Release 4 Message Queue will work fine with data backed up before the upgrade to Release 5.

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 4 to Release 5. 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 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Message Queue to Release 5 is the same as that for upgrading Release 4 Message Queue to Release 5.

To upgrade Release 3 Message Queue to Release 5, use the instructions in "Upgrading Message Queue from Java ES Release 4" on page 187, except substitute Release 3 wherever Release 4 is referenced.

Upgrading Message Queue from Java ES Release 2

This section includes information about upgrading Message Queue from Java ES 2004Q2 (Release 2) to Java ES Release 5. The section covers the following topics:

- Introduction
- Release 2 Compatibility Issues
- Release 2 Message Queue Upgrade
- Multiple Instance Upgrades

NOTE If you are upgrading from Release 2 Message Queue on the Linux platform, then you will have to perform a dual upgrade, in which both Message Queue *and* the operating system are upgraded (Release 5 Message Queue is not supported on RHEL 2.1). See "Dual Upgrade" on page 186 for more information.

Introduction

When upgrading Java ES Release 2 Message Queue to Release 5, 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 2 automatically.
- **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 5 is not supported. In particular, all Java ES shared components used by Message Queue need to be upgraded.

  In addition, Release 5 Message Queue is optionally dependent on Directory Server and Web Server (or Application Server), as described in "Message Queue Dependencies" on page 185. If these are installed on the same computer, upgrade of these components to Release 5 is also required.
- **Backward Compatibility.** Release 5 Message Queue is not fully compatible with Release 2, as described in "Release 2 Compatibility Issues," below.

- **Upgrade Rollback.** Rollback from Release 5 to Release 2 is not currently supported (see "Rolling Back the Upgrade").
- **Platform Issues.** The general approach for upgrading Message Queue is the same on both Solaris and Linux operating systems, however there are some additional procedures required on Linux. The procedures that follow indicate platform-specific commands, file locations, or procedures where appropriate.

Release 2 Compatibility Issues

Release 5 Message Queue introduces the following general Message Queue compatibility issues with respect to Release 2 and earlier versions.

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 191 and page 202.

Broker Compatibility

A Release 5 Message Queue broker will inter-operate with a Release 4, 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 5 Message Queue can use Release 4, Release 3, and Release 2 data, except that on Linux systems, Release 2 data must be first migrated to Release 5.

When updating to Release 5 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 5 Message Queue brokers.
- Any persistent Message Queue data—messages, destinations, durable subscriptions—is automatically converted, if necessary, to Release 5 Message Queue data when starting up a broker for the first time. For example, any existing destinations will be converted, if necessary, to Release 5 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 5 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 5 Message Queue administered objects are identical to Release 3 and Release 4 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 5, 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 5. The Administration Console (`imqadmin`) and the ObjectManager command line utility (`imqobjmgr`), when performing an update operation, will convert Release 2 administered objects into Release 5 administered objects.

- The Release 5 client runtime will look up and instantiate Release 2 administered objects and convert them for use by Release 5 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 5. However, if they are to use the *new* administered object attributes (see Chapter 16 of the *Message Queue 3.7 UR1 Administration Guide*, http://docs.sun.com/doc/819-4467 for information on administered object attributes), they will need to be rewritten. (Re-compiling Release 2 clients with Release 5 will show which Message Queue Release 2 attributes have been renamed in Release 5. 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 5. However, if they are to use the *new* administered object attributes (see Chapter 16 of the *Message Queue 3.7 UR1 Administration Guide*, http://docs.sun.com/doc/819-4467 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 5 administration tools (the Administration Console and command line utilities) only work with Release 3, Release 4, and Release 5 brokers. However, all Release 2 commands and command options remain supported.

Client Compatibility

Release 3 and Release 4 clients are completely compatible with Release 5 Message Queue. When upgrading from Release 2 to Release 5, however, you should consider the following compatibility issues, regarding Java clients:

- A Release 5 broker will support a Release 2 client (but without additional Release 5 capabilities).
- A Release 5 Java client can connect to a Release 2 broker (but without additional Release 5 capabilities).
- C client programs are supported by Release 5 and by Release 2, Release 3, or Release 4 brokers running with an Enterprise Edition license or a Platform Edition trial license.

Release 2 Message Queue Upgrade

This section describes how to perform a Message Queue upgrade from Java ES Release 2 to Java ES Release 5:

- Pre-Upgrade Tasks
- Upgrading Release 2 Message Queue (Solaris)
- Upgrading Release 2 Message Queue (Linux)
- Installing the Compatibility Package (Linux)
- Verifying the Message Queue Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Message Queue software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Message Queue Dependencies
- Back Up Message Queue

Verify Current Version Information

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 10-5 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`

Release 5	`Sun Java(tm) System Message Queue 3.7`
	`Version:  3.7 UR1`

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 5. Message Queue has hard upgrade dependencies on only a couple of shared components.

When upgrading Message Queue dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Message Queue.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63).

2. **Sun Cluster (soft upgrade dependency).** Instructions for upgrading Sun Cluster to Release 5 are provided in Chapter 3, "Sun Cluster Software" on page 75.

3. **Directory Server (soft upgrade dependency).** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, "Directory Server" on page 99.

4. **Java DB (soft upgrade dependency).** You need to perform a fresh install of Release 5 Java DB when upgrading Message Queue.

5. **Web Container Software (soft upgrade dependency).** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, "Web Server" on page 133 and Chapter 11, "Application Server" on page 205, respectively.

Back Up Message Queue

It is always a good practice to back up application data in a production environment before performing an upgrade. For Solaris OS, dynamic data is stored in the following directory:
`/var/imq/instances/`*instanceName*.

For other operating systems, see the *Message Queue 3.7 UR1 Administration Guide*, `http://docs.sun.com/doc/819-4467`.

Upgrading Release 2 Message Queue (Solaris)

The upgrade of Message Queue software to Java ES Release 5 makes use of the `mqupgrade` script, which installs Release 5 packages.

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 5 Message Queue.

4. Log in as Root.

   ```
   su -
   ```

5. Change directories to the location of the Tools directory of the Java ES Release 5 distribution.

   On Solaris SPARC:

   ```
   cd Solaris_sparc/Product/message_queue/Tools
   ```

   On Solaris x86:

   ```
   cd Solaris_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
   ```

Upgrading Release 2 Message Queue (Linux)

The upgrade of Release 2 Message Queue to Release 5 on the Linux platform is complicated by the fact that Java ES Release 2 is supported only on RHEL 2.1, but Java ES Release 5 is not supported on RHEL 2.1. Hence a dual upgrade is required: both the operating system and Message Queue must be upgraded. See "Dual Upgrades: Java ES and Operating System Softwared" on page 43

The basic procedure is to upgrade the Linux OS first, then upgrade all the Message Queue dependencies, and then upgrade Message Queue.

The upgrade from Release 2 Message Queue to Release 5 includes a data migration step that is not needed on Solaris systems, namely the migration of broker instance data to the appropriate Release 5 location. If you want to preserve your Release 2 data in upgrading to Release 5, Message Queue provides a migration tool, mqmigrate, to perform this migration.

To upgrade from Release 2 to Release 5, you use the same instructions as used in "Upgrading Release 2 Message Queue (Solaris)" on page 199, except you run the `mqmigrate` script (Step 6 on page 201) before you run the `mqupgrade` script (Step 7 on page 201), 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 5 Message Queue.

4. Log in as root or become superuser.

   ```
   su -
   ```

5. Change directories to the location `Tools` directory of the Java ES Release 5 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 (Linux)

If you have scripts or your Release 2 client applications contain scripts that depend on the location of Release 5 installed files, you will need to install the `sun-mq-compat` package, which contains symlinks from Release 2 file locations to Release 5 file locations.

The `sun-mq-compat` package is in the following location where you unzipped the Java ES Release 5 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-3.7-`*RelNo*`.i386.rpm`

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 are using the HTTP tunneling servlet to provide HTTP connection service support, the upgrade of Message Queue from Release 2 to Release 5 has upgraded the servlet. This requires you to re-deploy it after upgrading Message Queue to Release 5. See the *Message Queue 3.7 UR1 Administration Guide,* `http://docs.sun.com/doc/819-4467` for more information on HTTP support.

Also, you have to migrate Release 2 administered objects to their Release 5 versions using the Administration Console (`imqadmin`) and/or the ObjectManager command line utility (`imqobjmgr`) to perform an update operation.

Rolling Back the Upgrade

The upgrade of Message Queue from Release 2 to Release 5 is not currently supported. Normally the procedure would be similar to the rollback from Release 5 to Release 4 (see "Rolling Back the Upgrade" on page 191). However, because the upgrade of Message Queue from Release 2 to Release 5 does not update the Java ES product registry, the Java ES installer cannot re-install Release 2 Message Queue.

For work-arounds to this problem, please consult Sun Services.

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 2 to Release 5. 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.

Chapter 11

Application Server

This chapter describes how to upgrade Application Server to Java ES 5 (Release 5): Sun Java System Application Server Enterprise Edition 8.2.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Application Server Upgrades" on page 207
- "Upgrading Application Server from Java ES Release 4" on page 213
- "Upgrading Application Server from Java ES Release 3" on page 219
- "Upgrading Application Server from Java ES Release 2" on page 220
- "Upgrading the Solaris-bundled Application Server in a Solaris 10 Multi-zone Environment" on page 230

NOTE File locations in this chapter are specified with respect to directory paths referred to as *AppServer8-base* and *AppServer8Config-base* (Application Server 8.*x*), and *AppServer7-base* and *AppServer7Config-base* (Application Server 7.*x*). At least part of these paths might have been specified as installation directories or domain 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.

Table 11-1 Application Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
AppServer8-base	`/opt/SUNWappserver/appserver`	`/opt/sun/appserver`
AppServer8Install-base	`/opt/SUNWappserver`	`/opt/sun/appserver`

Table 11-1 Application Server Directory Paths *(Continued)*

Path Name Variable	Solaris OS	Linux OS
AppServer8Config-base	`/var/opt/SUNWappserver`	`/var/opt/sun/appserver`
AppServer7-base	`/opt/SUNWappserver7`	`/opt/SUNWappserver7`
AppServer7Config-base	`/var/opt/SUNWappserver7`	`/var/opt/SUNWappserver7`

NOTE The default Application Server domain name for Release 5 Application Server (8.*x)* is `domain1`. In other words, `domain1` is the default value of the *domainName* variable used in this chapter.

In addition, the default Directory Administration Service (DAS) instance name is `server`. J2EE Applications are not normally deployed to the DAS instance but to other standalone instances.

For more information regarding the `asadmin` commands used in this chapter, consult the *Sun Java System Application Server Enterprise Edition 8.2 Reference Manual*, `http://docs.sun.com/doc/819-4736`.

Overview of Application Server Upgrades

This section describes the following general aspects of Application Server that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5 Application Server
- Application Server Upgrade Roadmap
- Application Server Data
- Application Server Upgrade Strategy

About Java ES Release 5 Application Server

Java ES Release 5 Application Server represents a minor release with respect to Release 4, including only selected bug fixes. Release 5 Application Server is functionally the same as Release 4.

Application Server Upgrade Roadmap

There are two sets of upgrade paths that apply to upgrade of Application Server to Java ES Release 5:

- Table 11-2 shows the supported Java ES Application Server upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.
- Table 11-3 shows the supported Solaris-bundled Application Server upgrade paths to Java ES Release 5. Application Server Platform Edition is bundled with Solaris OS software. Upgrade of the bundled versions of Application Server to Release 5 Enterprise Edition can be performed using the Java ES installer, as indicated in Table 11-3.

Table 11-2 Upgrade Paths to Java ES 5 (Release 5): Application Server Enterprise Edition 8.2

Java ES Release	Application Server Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Application Server Enterprise Edition 8.1 2005Q4	Direct upgrade: Performed using Java ES installer.[1]	None
Release 3	Sun Java System Application Server Enterprise Edition 8.1 2005Q1	Direct upgrade: Performed using Java ES installer.	None
Release 2	Sun Java System Application Server 7.0 Update 3 (2004Q2) Platform and Enterprise Editions	Direct upgrade: Use the Java ES installer to upgrade packages, the `postInstall` script to perform reconfiguration, and the `asupgrade` utility to migrate domain information.[2]	Environment variables, domains, and other configuration data. J2EE components and applications need to be migrated to new Application Server environment and redeployed.
Release 1	Sun ONE Application Server 7.0 Update 1 (2003Q4) Platform and Enterprise Editions	Direct upgrade not certified: Use the same approach as in upgrading from Release 2 Application Server.	Environment variables, domains, 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 5.	

[1] If you wish to upgrade Application Server from Release 4 to Release 5 without retaining configuration or domains information, you can use the Java ES Release 4 uninstaller to uninstall Release 4 Application Server and then use the Java ES Release 5 installer to freshly install Release 5 Application Server. However, if Release 4 Application Server had been installed using the Configure Later option, then before uninstalling Release 4 Application Server, you must first create a *$HOME/.asadminprefs* file (where *$HOME* is the home directory for the user who installs and runs Application Server). The file has the following two lines:

```
AS_ADMIN_PASSWORD=password
AS_ADMIN_USER=admin
```

[2] Special care must be taken if you are upgrading from Release 2 Application Server on a Solaris platform because both the Solaris-bundled Application Server and the Java ES Release 2 Application Server coexist on your computer. See Table 11-3 and the note following it.

Table 11-3 Upgrade Paths for Application Server Versions Bundled with Solaris OS

Solaris OS Version	Application Server Version	General Approach	Reconfiguration Required
Solaris 10	Sun Java System Application Server Platform Edition 8.0.0_01	Direct upgrade: Performed using Java ES installer.	None.
Solaris 9	Sun Java System Application Server 7.0.0_03c	Direct upgrade: If Application Server *has been used* and domains have been created, then domain information needs to be migrated. Use the approach documented in upgrading from Release 2 Application Server, in which you use the Configure Later option of the Java ES installer, the `postInstall` script, and the `asupgrade` utility. If Application Server has *not* been used or has been used without creating domains, use the approach documented in upgrading from Release 2 Application Server, except use the Configure Now option of the Java ES installer and do not use the `postInstall` script or the `asupgrade` utility.	Environment variables, domains (if they have been created), and other configuration data. J2EE components and applications need to be migrated to new Application Server environment and redeployed.

NOTE Special care must be taken if you are upgrading from Release 2 Application Server on a Solaris platform because both the bundled version and the Java ES version of Application Server coexist on your computer. As a result you have to first uninstall the bundled version (and any corresponding domains) before proceeding with the upgrade from the Release 2 version. See "Solaris OS Only: Manually Remove the Application Server Packages Bundled with the Operating System" on page 222.

Application Server Data

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

Table 11-4 Application Server Data Usage

Type of Data	Location	Usage
Environment variables	*AppServer8-base*/config/asenv.conf	Global variables
Configuration data	Release 3, Release 4, & Release 5: 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, & Release 5: *AppServer8Config-base*/domains/*domainName*/applications Release 2: *AppServer7Config-base*/domains/*domainName*/*instanceName*/applications	Configuration of J2EE container for specific J2EE components and applications
Access log files	Release 3, Release 4, & Release 5: *AppServer8Config-base*/domains/*domainName*/logs/access/ Contains two files: server_acces_log and _asadmin_access_log Release 2: *AppServer7Config-base*/domains/*domainName*/*instanceName*/logs/access	Access logging

Application Server Upgrade Strategy

Your strategy for upgrading Application Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Application Server by presenting issues that might influence your Application Server upgrade plan.

Compatibility Issues

Release 5 Application Server does not introduce any interface changes with respect to Release 4 or Release 3. However, there are major interface changes between Release 5 and Release 2, making Release 5 incompatible with Release 2.

Release 5 Application Server, however, does not support Release 4 Service Registry. If Application Server is upgraded to Release 5, the Service Registry must also be upgraded to Release 5.

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 1-9 on page 47).
- **Message Queue.** Application Server depends on Message Queue to provide J2EE Java Message Service-compliant asynchronous messaging support.
- **High Availability Session Store.** Application Server depends upon High Availability Session Store (HADB) to maintain session state information needed to support failover between instances.

NOTE If you have an earlier installation of Application Server that does not require HADB, HADB must be installed before you can upgrade Application Server to Release 5. For example, Release 5 HADB must be installed to upgrade the Application Server versions bundled with Solaris OS. The Java ES installer will automatically perform the installation of HADB in these situations.

- Java DB**.** Application Server depends upon Java DB as the default developer database and to store sample application data and data required for Enterprise Java Beans timers.
- **Web Container (optional dependency).** 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).

Dual Upgrade

Dual upgrades, in which both Application Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up existing Application Server data.

   See "Application Server Data" on page 210 for the location of essential data.

2. Install the new operating system.

   The operating system installation can only be on the same computer and will wipe out the existing file system.

3. Restore the Application Server data that was backed up in Step 1.
4. Install Release 5 Application Server.

   Use the procedure documented in the relevant upgrade section of this chapter, depending on the version of Application Server data that was backed up in Step 1.

In-place Operating System Upgrade

1. Back up existing Application Server data.

   See "Application Server Data" on page 210 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Application Server.

   See the relevant section of this chapter, depending on upgrade path.

Upgrading Application Server from Java ES Release 4

This section includes information about upgrading Application Server from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Application Server Upgrade

Introduction

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

- **General Upgrade Approach.** The upgrade is performed using the Java ES installer. No reconfiguration of Application Server and no reconfiguration or migration of J2EE components is required to upgrade from Release 4 Application Server to Release 5.
- **Upgrade Dependencies.** Application Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Application Server. Application Server has a hard upgrade dependency only on the NSS shared component.

  In addition, as described in "Application Server Dependencies" on page 211, Release 5 Application Server has dependencies upon Message Queue, HADB, and Java DB. These are hard upgrade dependencies: all must be upgraded to Release 5.

  In addition, Application Server is optionally dependent on Java ES Web Server or third-party web containers. However, these are soft upgrade dependencies; upgrade of the web container is optional with respect to upgrade of Application Server to Release 5.
- **Backward Compatibility.** Release 5 Application Server is backwardly compatible with the Release 4 version.
- **Upgrade Rollback.** The upgrade to Release 5 cannot be rolled back to Release 4.
- **Platform Issues.** The general approach for upgrading Application Server is the same on both Solaris and Linux operating systems.

Release 4 Application Server Upgrade

This section describes how to perform an upgrade of Application Server from Java ES Release 4 to Java ES Release 5 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
- Upgrading Release 4 Application Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Application Server software you should perform the following tasks:

- Verify Current Version Information
- Upgrade Application Server Dependencies
- Back Up Application Server Data
- Obtain Required Configuration Information and Passwords

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 11-5 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 8.1`
Release 4	`Sun Java Enterprise System Application Server Enterprise Edition 8.1_02`
Release 5	`Sun Java Enterprise System Application Server Enterprise Edition 8.2`

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 5. Application Server has hard upgrade dependencies on the NSS shared component and on Message Queue, HADB, and Java DB product components.

When upgrading Application Server dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Application Server. However, upgrade of shared components as well as Message Queue, HADB, and Java DB is normally achieved automatically by the Java ES installer when upgrading Application Server.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Application Server are upgraded automatically by the Java ES installer when you perform an upgrade of Application Server to Release 5.

2. **Message Queue.** Instructions for upgrading Message Queue to Release 5 are provided in Chapter 10, "Message Queue" on page 181.

3. **High Availability Session Store (HADB).** Instructions for upgrading HADB are provided in Chapter 9, "High Availability Session Store" on page 169.

4. **Java DB.** Instructions for upgrading Java DB are provided in Chapter 8, "Java DB" on page 159.

5. **Web Container Software (soft upgrade dependency).** Instructions for upgrading Web Server are provided in Chapter 7, "Web Server" on page 133.

Back Up Application Server Data

The Application Server upgrade from Release 4 to Release 5 does not modify configuration data. There is therefore 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 4 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 5 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 during upgrade of the installed image.
- In multiple node deployments, perform the upgrade procedure on each node or computer that hosts Application Server instances.

Upgrade Procedure

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

1. Log in as root or become superuser.

   ```
   su -
   ```

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.

   a. Stop all running node agents.

      ```
      AppServer8-base/bin/asadmin stop-node-agent --user admin_ID
             nodeagentName
      ```

      where *nodeagentName* has the form *hostName_domainName,* but is simply *hostName* by default.

   b. Stop the Domain Administration Server (DAS).

      ```
      AppServer8-base/bin/asadmin stop-domain --user admin_ID
             domainName
      ```

c. Stop the PointBase database server (if being used).

```
AppServer8Config-base/appserver/pointbase/tools/stopserver.sh
```

4. Launch the Java ES installer.

```
cd Java ES Release 5 distribution/os_arch
./installer
```

where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

5. Select Application Server in the component selection page.

As hard upgrade dependencies, Message Queue, HADB, and Java DB will also be automatically selected for upgrade.

6. Choose to Configure Now.

7. Specify the configuration values requested.

You will be presented with a number of configuration panels.

8. Confirm your upgrade choice.

Application Server packages (and, if necessary, those for Message Queue, HADB, and Java DB) will be upgraded and an upgrade summary displayed.

9. Exit the Java ES installer.

10. Restart the upgraded Domain Administration Server (DAS).

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
     domainName
```

11. Restart the upgraded Application Server instances.

```
AppServer8-base/bin/asadmin start-node-agent --user admin_ID
     nodeagentName
```

where *nodeagentName* has the form *hostName_domainName*, but is simply *hostName* by default.

Verifying the Upgrade

You can verify successful upgrade using the following command:

```
AppServer8-base/bin/asadmin version --verbose
```

See Table 11-5 on page 214 for output values.

Post-Upgrade Tasks

If you want to continue using the embedded Release 4 PointBase database in stead of Java DB, a new Release 5 Java ES product component, you have to manually edit the upgraded *AppServer8-base*`/config/asenv.conf` file.

After upgrading Application Server from Release 4, the PointBase settings in `asenv.conf` are as follows:

```
AS_POINTBASE="%POINTBASE_HOME%"
AS_POINTBASE_SAMPLESDB="%POINTBASE_SAMPLESDB%"
```

Change these settings to the following values:

```
AS_POINTBASE="AppServer8-base/pointbase"
AS_POINTBASE_SAMPLESDB="AppServerConfig8-base/var/appserver/pointbase"
```

Rolling Back the Upgrade

Rollback of the Release 5 upgrade is not supported.

Upgrading Application Server from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Application Server to Release 5 is the same as that for upgrading Release 4 Application Server to Release 5.

To upgrade Release 3 Application Server to Release 5, use the instructions in "Upgrading Application Server from Java ES Release 4" on page 213, except substitute Release 3 wherever Release 4 is referenced.

Upgrading Application Server from Java ES Release 2

This section includes information about upgrading Application Server from Java ES Release 2 to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 2 Application Server Upgrade

NOTE If you are upgrading from Release 2 Application Server on the Linux platform, then you will have to perform a dual upgrade, in which both Application Server *and* the operating system are upgraded (Release 5 Application Server is not supported on RHEL 2.1). See "Dual Upgrade" on page 212 for more information.

Introduction

When upgrading Java ES Release 2 Application Server to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by installing Release 5 Application Server using the Java ES installer and choosing the configure later option. Reconfiguration is subsequently achieved using the `asupgrade` utility. Following the Application Server upgrade you have to migrate Release 2 J2EE components and applications to Release 5.
- **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 5 is not supported.

  In particular, all Java ES shared components required by Application Server must be upgraded. Message Queue, if residing on the same computer, must also be upgraded. High Availability Session Store (HADB) and Java DB must be installed, and if Web Server is being used for load balancing, it also must be upgraded.
- **Backward Compatibility.** Release 5 Application Server is not backwardly compatible with the Release 2 version. J2EE components and applications need to be migrated run in a Release 5 Application Server environment.

- **Upgrade Rollback.** Rollback of the Release 5 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 approach for upgrading Application Server on the Solaris OS cannot be directly applied to the Linux OS: Java ES Release 2 is only supported on Linux2.1, while Java ES Release 5 is not supported on Linux2.1. Hence a dual upgrade is required: both the operating system and Application Server must be upgraded (see "Dual Upgrades: Java ES and Operating System Softwared" on page 43)
- **Data Locations.** There are new Release 5 locations of configuration data, deployment data, and access log files, as specified in Table 11-4 on page 210.

Release 2 Application Server Upgrade

This section describes how to perform an upgrade of Application Server from Java ES Release 2 to Java ES Release 5 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
- Upgrading Release 2 Application Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Application Server you should perform the following tasks:

- Solaris OS Only: Manually Remove the Application Server Packages Bundled with the Operating System
- Verify Current Version Information
- Upgrade Application Server Dependencies
- Back Up Application Server Data
- Obtain Required Configuration Information and Passwords

Solaris OS Only: Manually Remove the Application Server Packages Bundled with the Operating System

The installation of Release 2 Application Server did not remove the version of Application Server bundled with the Solaris OS. To properly upgrade Release 2 Application Server to Release 5, you have to perform the following steps.

1. Uninstall the bundled version of by manually removing the corresponding Application Server packages located in the `/usr/appserver` directory:

   ```
   pkgrm SUNWascmnse SUNWaslb SUNWasut ...
   ```

   where the full set of packages can be obtained using the following command:

   ```
   pkginfo -i|grep -i "application server"
   ```

   The results would include packages such as:

   `SUNWasacee, SUNWascmnse, SUNWasu, SUNWasuee, SUNWasac, SUNWascmn, SUNWasdb, SUNWasdem, SUNWasdemdb, SUNWasr, SUNWasut, SUNWasman, SUNWasjdoc`

   and might include localization packages as well:

   `SUNW`*Locale*`asacee`, `SUNW`*Locale*`ascmnse`, `SUNW`*Locale*`asu`, `SUNW`*Locale*`asuee`

2. If the bundled version has been used, delete the domains information:

   ```
   rm /usr/appserver/domains
   ```

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 11-5 on page 214 for version outputs.

Upgrade Application Server Dependencies

The upgrade of Application Server dependencies requires the upgrade to Release 5 of all locally-resident Release 2 components upon which Application Server depends, specifically Message Queue and shared components. Shared components, however, are upgraded automatically by the Java ES installer as part of the upgrade procedure (see "Upgrade Procedure" on page 224).

In addition, the upgrade of Application Server requires that HADB, and Java DB, upon which Release 5 Application Server depends, be installed. These product components, however, are automatically selected and installed by the Java ES installer as part of the upgrade procedure (see "Upgrade Procedure" on page 224).

In general, when upgrading Application Server dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Application Server. However, upgrade of shared components as well as installation of HADB,and Java DB are performed automatically by the Java ES installer when upgrading Application Server.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Application Server are upgraded automatically by the Java ES installer when you perform an upgrade of Application Server to Release 5.

   NOTE Upgrade of shared components to Release 5 includes the upgrade of J2SE to JDK version 1.5, which is not supported by Release 2 Application Server. Hence, before upgrading shared components, you should shut down Release 2 Application Server. Shared components are upgraded to release 5 whenever the Java ES installer is used to install or upgrade any Java ES product component (such as Directory Server, Message Queue, HADB, Java DB, and others).

2. **Message Queue.** See "Upgrading Message Queue from Java ES Release 2" on page 194. Message Queue cannot be upgraded from Release 2 using the Java ES installer.

3. **High Availability Session Store (HADB)**. A fresh install of HADB is performed by the Java ES installer when upgrading Application Server.

4. **Java DB**. A fresh install of Java DB is performed by the Java ES installer when upgrading Application Server.

5. **Web Container Software (soft upgrade dependency).** See "Upgrading Web Server from Java ES Release 2" on page 157.

Back Up Application Server Data

The Application Server upgrade from Release 2 to Release 5 does not overwrite Release 2 configuration data. However, for safe measure, the entire `domains` directory should be backed up before performing the upgrade to Release 5:

AppServer7Config-base`/domains`

Obtain Required Configuration Information and Passwords

You should know the following information about your currently installed version:

- Application Server administrator user ID and 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 5 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. If Java ES shared components have been upgraded to Release 5 (in particular J2SE upgraded to JDK 1.5) while Release 2 Application Server was still running, then edit the `asenv.conf` file to point directly to JDK 1.4.

   ```
   AS_JAVA=/usr/java
   ```

   Application Server cannot be shut down (Step 3, below) if `asenv.conf` is referencing JDK 1.5.

3. Stop all Application Server and related processes.

   AppServer7-base`/bin/asadmin stop-appserv` *domainName*

4. Install Release 5 Application Server.

Perform the following steps:

a. Launch the Java ES installer on the computer hosting Release 2 Application Server.

```
cd Java ES Release 5 distribution/os_arch
./installer
```

where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.") Application Server is *not* shown as upgradable from Release 2.

b. Select Application Server from the component selection page.

Choose to install at least the first three subcomponents, including the Node Agent component.

c. Click Next.

If Message Queue has not already been upgraded to Release 5, an error message window asks you to upgrade Message Queue. In that case, click OK and upgrade Message Queue. The procedure is documented in "Release 2 Message Queue Upgrade" on page 197.

d. Specify an installation directory different from that in which Release 2 Application Server was installed.

e. Select the Configure Later option.

NOTE	If you are the version of Application Server bundled with Solaris 9 OS, and have never used this version (no domain information has been created), then you can select Configure Now and omit Step 5 and succeeding steps.

f. If needed, select the option to install localized packages.

g. Exit the Java ES installer when installation is complete.

5. Perform the following post-install procedure:

   a. Locate the `postInstall` readme file in the Application Server tools directory of the Java ES Release 5 distribution:

      Java ES Release 5 distribution`/`*os_arch*`/Product/application_svr/Tools`

      where *os_arch* matches your platform, such as `Solaris_sparc`.

   b. Refer to the `ReadMe` file and run the `postInstall` script.

      ```
      cd Java ES Release 5 distribution/os_arch/Product/application_svr/Tools
      ./postInstall AppServer8Install-base AppServer8Config-base
      ```

      The scripts configure and create the *AppServer8-base*`/bin/*` shell scripts and a `config/asenv.conf` file from templates that are installed during installation. (Normally the Java ES installer creates the `bin/*` shell scripts, but if Configure Later is chosen, they have to be created as described.)

      NOTE When you upgrade Application Server to Release 5, non-admin ports 8080 (default) and 8181 (secure) are allocated to the Domain Administration Server (DAS). These port assignments could possibly conflict with the ports allocated to Release 2 Application Server instances. If this is the case, then you must change the DAS ports to avoid the conflict. For instructions, see the section on how to modify http-listener attributes in the *Sun Java System Application Server Enterprise 8.2 Edition Administration Guide*, `http://docs.sun.com/doc/819-4733`.

   c. If necessary, modify the environment settings in the *AppServer8-base*`/config/asenv.conf` 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 8.2 Edition High Availability Administration Guide*, `http://docs.sun.com/doc/819-4740`.

6. Run the asupgrade utility.

   The asupgrade utility creates a Release 5 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.

   a. Identify both target and source directories for the migration of domains information:

      - Release 2: *AppServer7-base*

      - Release 5: *AppServerConfig8-base*/domains

   b. Provide the admin user, admin password, and master password of Release 2 Application Server.

      If asked for a master password, specify changeit, and if you are using a keystore that has a master password other than changeit you should change the password to changeit.

      ```
      jdk-home/bin/keytool -storepasswd -new changeit -keystore keystore
      -storepass oldpasswd
      ```

   For more information about the Application Server asupgrade utility, refer to the user commands section of the *Sun Java System Application Server Enterprise Edition 8.2 Reference Manual*, http://docs.sun.com/doc/819-4736. and the *Sun Java System Application Server Enterprise Edition 8.2 Upgrade and Migration Guide*, http://docs.sun.com/doc/819-4737/6n6sao3ju?a=view.

7. If you had redirected the asenv.conf file in Step 2 on page 224, restore it to point to JDK 1.5.

   ```
   AS_JAVA=/usr/jdk/entsys-j2se
   ```

   Application Server cannot be started (Step 8 and Step 9, below) if asenv.conf is referencing JDK 1.4.

8. Start the Domain Administration Server (DAS).

   ```
   AppServer8-base/bin/asadmin start-domain --user admin_ID
       domainName
   ```

9. 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
     nodeagentName
```

where *nodeagentName* has the form *hostName_domainName*, but is simply *hostName* by default.

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 Server to Release 5 by entering the following command:

```
AppServer8-base/bin/asadmin version --verbose
```

See Table 11-5 on page 214 for output values.

Post-Upgrade Tasks

There are two post-upgrade tasks you have to perform beyond the steps described in "Upgrade Procedure" on page 224:

- Correct the Reference to JSS
- Migrate Release 2 J2EE Components

Correct the Reference to JSS

For Application Server to reference the correct version of the JSS shared component, you have to replace `jss3.jar` with `jss4.jar` in the Application Server `domain.xml` file located at:

```
AppServer8Config-base/domains/domainName/config/domain.xml
```

Migrate Release 2 J2EE Components

You have to migrate Release 2 J2EE components and applications to run in a Release 5 Application Server environment and redeploy 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 8.2 Upgrade and Migration Guide*, `http://docs.sun.com/doc/819-4737`.

Rolling Back the Upgrade

The procedure for rolling back the upgrade to Release 5 of Application Server is simply to revert to the Release 2 version of Application Server, which was not removed by the upgrade to Release 5.

Upgrading the Solaris-bundled Application Server in a Solaris 10 Multi-zone Environment

Application Server (as well as Message Queue, upon which Application Server depends), is bundled with the Solaris 10 OS. Unless removed from the global zone, the bundled Application Server is propagated to non-global zones when such zones are created. The existence of the bundled Application Server in all zones impacts the subsequent upgrade of Application Server to Release 5:

- Upgrading Application Server in the global zone, removes the bundled version in the global zone and automatically removes the bundled version from all non-global zones.
- Upgrading Application Server to Release 5 in the global zone does not upgrade Application Server in non-global zones because Release 5 Application Server packages are not propagated.

The following example is provided to document some of the subtleties involved in upgrading Application Server in a Solaris 10 multi-zone environment. (For a more comprehensive discussion regarding Java ES and Solaris 10 zones, see "Java ES 5 Upgrade and Solaris 10 Zones" on page 58.)

The objective in the example is to upgrade to Release 5 the Solaris-bundled Application Server (version 8.0.0_01) in a Solaris 10 sparse root zone.

You cannot simply upgrade Application Server in a sparse root zone because the Solaris-bundled Application Server is installed in a read-only directory mounted from the global zone. Hence, to upgrade Application Server to Release 5 in the sparse root zone, you must first remove the bundled version in the global zone.

In addition, Message Queue is installed in the global zone, representing a departure from the practice by which only shared components (not product components) are to be installed in the global zone. This is because Message Queue cannot be installed or upgraded in a sparse root zone because of the read-only directories.

The procedure for upgrading the Solaris-bundled Application Server (version 8.0.0_01) in a Solaris 10 sparse root zone to Release 5 is as follows:

1. Verify the initial state of your system.

   This example assumes a version of Solaris 10 with a sparse root zone that has been configured, installed, and booted by the global administrator.

The sparse root zone includes all Java ES components that are already installed in the global zone, namely the versions of Message Queue and Application Server bundled with Solaris 10.

In addition, the example assumes that the user has previously used the bundled Application Server in the sparse root zone, having created administrative domain information that needs to be preserved.

2. Upgrade the bundled version of Application Server in the global zone.

   This operation removes the bundled Application Server packages and replaces them with Release 5 packages. The removal of the bundled packages is propagated to the sparse root zone, effectively uninstalling Application Server packages in the sparse root zone, but the Release 5 packages are not propagated to non-global zones.

   a. Run the Java ES installer in the global zone.

   b. Select Application Server in the component selection page.

      Message Queue, HADB, and Java DB will automatically be selected, and Application Server and Message Queue will be marked as upgradable.

   c. Complete the upgrade.

      While Release 5 Message Queue will propagate to the sparse root zone, neither Application Server, HADB, nor Java DB will be propagated. In addition all shared components will be synchronized to Release 5 and propagated to the sparse root zone.

3. Install Application Server in the sparse root zone.

   a. Run the Java ES installer in the sparse root zone.

   b. Select Application Server in the component selection page.

      De-select Message Queue if it is automatically selected, and select HADB and Java DB if they are not automatically selected.

   c. Complete the installation of Application Server.

      Choose to Configure Later so that domain information is not overwritten by the installation process.

      In the case where no domains had been created, you can choose to Configure Now.c

Chapter 12

Service Registry

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

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Service Registry Upgrades" on page 234
- "Upgrading Service Registry from Java ES Release 4" on page 237

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

The default values of these directory paths are shown in the following table.

Table 12-1 Service Registry Directory Paths

Path Name Variable	Solaris OS	Linux OS
ServiceRegistryR4-base	`/opt/SUNWsoar`	`/opt/sun/SUNWsoar`
RegistryDomainR4-base	`/var/opt/SUNWsoar`	`/var/opt/sun/SUNWsoar`
ServiceRegistryR5-base	`/opt/SUNWsrvc-registry`	`/opt/sun/srvc-registry`
RegistryDomainR5-base	`/var/opt/SUNWsrvc-registry`	`/var/opt/sun/srvc-registry`

Overview of Service Registry Upgrades

This section describes the following general aspects of Service Registry that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5
- Java ES Release 5 Upgrade Roadmap
- Service Registry Data
- Service Registry Upgrade Strategy

About Java ES Release 5

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

Java ES Release 5 Upgrade Roadmap

Table 12-2 shows the supported Service Registry upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 12-2 Upgrade Paths to Java ES 5 (Release 5): Service Registry 3.1

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

Service Registry Data

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

Table 12-3 Service Registry Data Usage

Type of Data	Location	Usage
Installation parameters	*ServiceRegistryR4-base*`/install/install.properties`	Configuration of Service Registry
Trusted certificates	*ServiceRegistryR4-base*`/install/cacerts`	Certificates trusted by Service Registry that are not part of Application Server installation
Configuration data	*RegistryDomainR4-base*`/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes/*.properties`	Configuration of Service Registry instance
Registry/repository data	*RegistryDomainR4-base*`/3.0/data`	Database and user certificates
Web interface configuration	*RegistryDomainR4-base*`/3.0/jaxr-ebxml`	Configuration of web interface

Service Registry Upgrade Strategy

Your strategy for upgrading Service Registry generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Service Registry by presenting issues that might influence your Service Registry upgrade plan.

Compatibility Issues

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

Dependencies

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

Service Registry has dependencies on the following Java ES components:

- **Shared components.** Service Registry has dependencies on specific Java ES shared components (see Table 1-9 on page 47).
- **Application Server.** Service Registry has a mandatory dependency on Application Server to provide a container for the Service Registry application and, in Java ES Release 5, to manage connections to the networked registry/repository database.
- **Java DB.** Service Registry has a mandatory dependency on Java DB as the default database for storing services and the meta data describing them.

Dual Upgrade

Dual upgrades, in which both Service Registry and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed in either of two ways:

- Fresh operating system installation
- In-place operating system upgrade

Fresh Operating System Installation

1. Back up existing Service Registry data.

   See "Service Registry Data" on page 235 for the location of essential data.

2. Install the new operating system.

   The operating system installation can be on a new system (or a Solaris 10 zone) or it can wipe out the existing file system.

3. Restore the Service Registry data that was backed up in Step 1.
4. Install Release 5 Service Registry.

In-place Operating System Upgrade

1. Back up existing Service Registry data.

   See "Service Registry Data" on page 235 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Service Registry.

   See the "Upgrading Service Registry from Java ES Release 4" on page 237.

Upgrading Service Registry from Java ES Release 4

This section includes information about upgrading Service Registry from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Service Registry Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is achieved by performing a fresh install of Release 5 Service Registry, migrating the Release 4 data and configuration to Release 5, and then removing Release 4 to conserve disk space.
- **Upgrade Dependencies.** Service Registry has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Service Registry.

  Service Registry has hard upgrade dependencies on Application Server and Java DB.
- **Backward Compatibility.** Release 5 Service Registry is fully compatible with Release 4.
- **Upgrade Rollback.** A rollback of the Release 5 upgrade is achieved by reverting to Release 4 after restoring the saved database and configuration data.
- **Platform Issues.** The general approach for upgrading Service Registry is the same on both Solaris and Linux operating systems.

Release 4 Service Registry Upgrade

This section describes how to perform an upgrade of Service Registry from Java ES Release 4 to Java ES Release 5 on both the Solaris and Linux platforms. 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
- Upgrading Release 4 Service Registry
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Service Registry, you should perform the following tasks:

- Verify Current Version Information
- Upgrade Service Registry Dependencies
- Back Up Service Registry Data
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Service Registry by observing the characteristics of the Web Console user interface:

```
http://localhost:6060/soar
```

Also, you can check Service Registry package names. For example:

On Solaris:

```
pkginfo -l|grep srvc
```

On Linux:

```
rpm -qa|grep srvc
```

The distinguishing characteristics and package names are shown in the following table:

Table 12-4 Service Registry Version Verification Outputs

Java ES Release	Service Registry Version Number	Distinguishing Characteristic
Release 4	`3.0`	Web Console: tools section in left-hand panel Package names include the string: `soar`
Release 5	`3.1`	Web Console: three tabs in left-hand panel Package names include the string: `srvc-registry`

Upgrade Service Registry 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 5. Service Registry has hard upgrade dependencies on a number of shared components, Application Server, and Java DB.

When upgrading Service Registry dependencies, you should do so in the order below (skipping any that might already have been upgraded), before you upgrade Service Registry. Upgrade of shared components is normally achieved automatically by the Java ES installer.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Service Registry are upgraded automatically by the Java ES installer when you perform an upgrade of Service Registry to Release 5.

2. **Java DB.** Instructions for upgrading Java DB to Release 5 are provided in Chapter 8, "Java DB" on page 159.

3. **Application Server.** Instructions for upgrading Application Server to Release 5 are provided in Chapter 11, "Application Server" on page 205.

Modify the HTTP Port Number

Edit the *ServiceRegistryR4-base*`/install/install.properties` file to change the HTTP port from `6060` to `6480` (`6060` is a reserved port). For information on setting this property, see the *Service Registry 3.1 Administration Guide,* `http://docs.sun.com/doc/819-4640`.

Back Up Service Registry Data

The Service Registry upgrade from Release 4 to Release 5 does not modify configuration data or the registry/repository database. There is no need to back up current data.

Obtain Required Configuration Information and Passwords

You need to know the user IDs, passwords, domain name, and port number for your Release 4 Service Registry.

Upgrading Release 4 Service Registry

This section describes the upgrade procedure on Solaris and Linux platforms.

Upgrade Procedure (Solaris)

The procedure documented below applies to Service Registry instances residing locally on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Make sure that the Jakarta ANT Java/XML-based build tool (ANT shared component) references the correct version of J2SE.

   (The `ant` command is used in the steps that follow.)

   ```
   PATH=/usr/jdk/entsys-j2se/bin:$PATH
   ```

   ```
   export PATH
   ```

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

   ```
   cd ServiceRegistryR4-base/install
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
   ```

   The domain is associated with a Service Registry instance.

4. Perform a fresh install of Release 5 Service Registry.

   Perform the following steps:

a. Launch the Java ES installer on the computer hosting Release 4 Service Registry.

```
cd Java ES Release 5 distribution/os_arch
./installer
```

where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

b. Select Service Registry from the component selection page.

c. Specify an installation directory path different from that of Release 4.

By default, the Release 5 installation path (*ServiceRegistryR5-base*) is different from the Release 4 installation path (*ServiceRegistryR4-base*).

d. Select the Configure Later option.

Configure Now is not supported.

e. If needed, select the option to install localized packages.

f. Exit the Java ES installer when installation is complete.

5. Upgrade and configure the Release 5 Service Registry instance.

```
cd ServiceRegistryR5-base/install
/usr/sfw/bin/ant -f build-install.xml
    -Dinstall.properties=ServiceRegistryR4-base/install/install.properties
    upgrade
```

As an alternative to pointing to the Release 4 `install.properties` file, you can modify the default Release 5 `install.properties` file to reproduce any Release 4 property values. For information on setting these properties, see the *Service Registry 3.1 Administration Guide*, http://docs.sun.com/doc/819-4640.

If you are using custom property values, but not putting them in `install.properties`, then you need to specify such property values on the Ant command line (all on one line), as follows:

```
/usr/sfw/bin/ant -f build-install.xml
    -Dregistry.install.RegistryServerKeystorePassword=passwd1
    -Dregistry.install.AdministratorPassword=passwd2
    -Dregistry.install.ApplicationServerKeystorePassword=passwd3
    upgrade
```

However, it is recommended that you include such custom property values in the `install.properties` file with restricted permissions to avoid the use of command-line settings that can be viewed by unauthorized personnel. See the Service Registry *Administration Guide* for more information.

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

6. If the server property files of the Release 4 Service Registry have been modified, you can make corresponding changes to the Release 5 Service Registry configuration as follows:

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

      (The domain was automatically started by the `upgrade` command of Step 5.)

      ```
      cd ServiceRegistryR5-base/install
      /usr/sfw/bin/ant -f build-install.xml appserver.domain.stop
      ```

   b. Transfer the Release 4 Service Registry instance configuration to Release 5.

      Add any modifications that you had made to the Release 4 Service Registry instance configuration:

      RegistryDomainR4-base`/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes/*.properties`

      to the corresponding Release 5 configuration:

      RegistryDomainR5-base`/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes/*.properties`

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

   ```
   cd ServiceRegistryR5-base/install
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.start
   ```

Upgrade Procedure (Linux)

Upgrading Service Registry on Linux is identical to Solaris (see "Upgrade Procedure (Solaris)" on page 240) except that the location of the `ant` command on the Linux platform, which is used in various steps of the upgrade procedure, is different from the location on Solaris platforms:

```
/opt/sun/share/bin/ant
```

Verifying the Upgrade

You can verify successful upgrade of Service Registry by observing the characteristics of the Web Console user interface:

```
http://localhost:6480/soar
```

Also, you can check Service Registry package names. For example:

On Solaris:
```
pkginfo -l|grep soar
```

On Linux:
```
rpm -qa|grep soar
```

The distinguishing characteristics and package names are shown in Table 12-4 on page 239.

Post-Upgrade Tasks

The following steps, which describe how to remove Release 4 Service Registry, should not be performed until you are certain you do not want to roll back the upgrade to Release 4.

1. Delete the Release 4 Service Registry (Application Server) domain:

   `cd` *ServiceRegistryR4-base*`/install`

   On Solaris:
   ```
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
   ```

   On Linux:
   ```
   /opt/sun/bin/ant -f build-install.xml appserver.domain.delete
   ```

2. Delete the directory containing the Release 4 Service Registry domain files.

   rm -rf *RegistryDomainR4-base*

3. Delete the directory containing the Release 4 Service Registry installation files.

   rm -rf *ServiceRegistryR4-base*

Rolling Back the Upgrade

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

1. Stop and delete the Release 5 Service Registry (Application Server) domain:

   `cd` *ServiceRegistryR4-base*`/install`

   On Solaris:
   ```
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.delete
   ```

   On Linux:
   ```
   /opt/sun/bin/ant -f build-install.xml appserver.domain.delete
   ```

2. Run the Java ES Release 5 uninstaller to uninstall Release 5 Service Registry.

3. Start the Release 4 Service Registry domain.

   `cd` *ServiceRegistryR4-base*`/install`

   On Solaris:
   ```
   /usr/sfw/bin/ant -f build-install.xml appserver.domain.start
   ```

   On Linux:
   ```
   /opt/sun/bin/ant -f build-install.xml appserver.domain.start
   ```

4. Access the Release 4 Service Registry Web Console.

   ```
   http://localhost:6480/soar
   ```

5. Confirm that the Console displays Release 4 characteristics as shown in "Service Registry Version Verification Outputs" on page 239.

Multiple Instance Upgrades

In some deployment architectures Service Registry is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Service Registry instances running on multiple computers with a load balancer to distribute the load.

In these architectures the registries are predominantly read-only and respond to a heavy query load by accessing a common database.

You perform the upgrade of Service Registry on each computer as described in "Release 4 Service Registry Upgrade" on page 238.

Chapter 13

Web Proxy Server

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

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. 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 *WebProxyServer-base*. At least part of this path might have been specified as an installation directory when Web Proxy Server was initially installed. If not, the Java ES installer assigns a default value.

The default value of *WebProxyServer-base* depends on operating system platform:, as shown in the following table.

Table 13-1 Web Proxy Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
WebProxyServer-base	`/opt/SUNWproxy`	`/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 5 (Release 5):

- About Java ES Release 5 Web Proxy Server
- Web Proxy Server Upgrade Roadmap
- Web Proxy Server Data
- Web Proxy Server Upgrade Strategy

About Java ES Release 5 Web Proxy Server

Java ES Release 5 Web Proxy Server represents a minor bug-fix release with respect to Release 4.

However, Release 5 Web Proxy Server includes better performance, more scalable architecture, better standards compliance, and a new administration interface as compared to Sun One Web Proxy Server 3.6, before its inclusion in Java Enterprise System.

Web Proxy Server Upgrade Roadmap

Table 13-2 shows the Web Proxy Server upgrade path to Java ES Release 5. Web Proxy Server was not included in previous Java ES releases. The table applies to the Solaris OS only, because Web Proxy Server was not previously supported on the Linux OS.

Table 13-2 Upgrade Paths to Java ES 5 (Release 5): Web Proxy Server 4.0.4

Java ES Release	Web Proxy Server Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Web Proxy Server 4.0.1 2005Q4	Direct upgrade: Performed using patches	None
Pre-dates Java ES releases (Solaris OS only)	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 13-3 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

Web Proxy Server Upgrade Strategy

Your strategy for upgrading Web Proxy Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Web Proxy Server by presenting issues that might influence your Web Proxy Server upgrade plan.

Compatibility Issues

Release 5 Web Proxy Server does not introduce any new public interfaces and is backwardly compatible with Release 4 Web Proxy Server. Release 5 Web Proxy Server is also compatible with 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 5.

Web Proxy Server Dependencies

Web Proxy Server has dependencies on the following Java ES components:

- **Shared components.** Web Proxy Server has dependencies on specific Java ES shared components (see Table 1-9 on page 47).
- **Directory Server.** Web Proxy Server has an optional dependency on Directory Server for providing LDAP-based authentication.
- **Web Server.** Web Proxy Server has a co-dependency on Web Server for providing improved security and performance for HTTP requests.

Dual Upgrade

Dual upgrades, in which both Web Proxy Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed using the in-place operating system upgrade approach:

1. Back up existing Web Proxy Server data.

   See "Web Proxy Server Data" on page 247 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Web Proxy Server.

   See the appropriate section of this chapter, depending on upgrade path.

Upgrading Web Proxy Server from Java ES Release 4

This section includes information about upgrading Web Proxy Server from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Release 4 Web Proxy Server Upgrade

Introduction

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

- **General Upgrade Approach.** The upgrade is performed using patches. There is no additional reconfiguration required.
- **Upgrade Dependencies.** Web Proxy Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), however Web Proxy Server has hard upgrade dependencies only on NSS and NSPR shared components.
- **Backward Compatibility.** Release 5 Web Proxy Server is backwardly compatible with Release 4.
- **Upgrade Rollback.** Rollback of the Release 5 upgrade of Web Proxy Server is achieved by removing the upgrade patches.
- **Platform Issues.** The general approach for upgrading Web 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 4 Web Proxy Server Upgrade

This section provides an overview of how to perform an upgrade of Web Proxy Server to Java ES Release 5. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading Release 4 Web Proxy Server (Solaris)
- Upgrading Release 4 Web Proxy Server (Linux)
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade (Solaris)

Pre-Upgrade Tasks

Before you upgrade Web Proxy Server, you should perform the following tasks:

- Verify Current Version Information
- Upgrade Web Proxy Server Dependencies
- Back Up Web Proxy Server Data
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Web Proxy Server by entering the following command:

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

Table 13-4 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`
Release 5	`4.0.4`

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 5. However, the upgrade of Web Proxy Server to Release 5 only requires that the NSS and NSPR shared components be upgraded. If these shared components have not yet been upgraded, you should synchronize all shared components to their Release 5 versions using the Synchronize Shared Components option. Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

Back Up Web Proxy Server Data

The Web Proxy Server upgrade to Release 5 does not modify Release 4 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 4 Web Proxy Server (Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Web Proxy Server software to Java ES Release 5 takes into account the following considerations:

- All Web Proxy Server instances corresponding to the same installed Web Proxy Server image are upgraded at the same time. All such instances should be shut down when patches are being applied to the installed image.
- The Release 5 Web Proxy Server upgrade patches for Solaris OS are shown in the following table:

Table 13-5 Patches[1] to Upgrade Web Proxy Server on Solaris

Description	Patch ID: SPARC Solaris 9 & 10	Patch ID: X86 Solaris 9 & 10
Web Proxy Server core	120981-10	120982-10
Web Proxy Server localization	122963-01	122964-01

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. 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 Web Proxy Server on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

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

   ```
   WebProxyServer-base/proxy-instanceName/stop
   WebProxyServer-base/proxy-admserv/stop
   ```

3. If you have not already done so, synchronize all shared component to Release 5.

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

4. Obtain the required patches, based on Table 13-5.

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

5. Apply the appropriate Web Proxy Server core and, if needed, localization patches in Table 13-5, in that order.

   ```
   patchadd /tmp/patch_ID
   ```

6. Confirm that the patch upgrades were successful:

   ```
   showrev -p | grep proxy
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Restart the Web Proxy Server instances that were stopped in Step 2.

   ```
   WebProxyServer-base/proxy-instanceName/start
   ```

Upgrading Release 4 Web Proxy Server (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Web Proxy Server software to Java ES Release 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations" on page 258), except that the Linux Release 5 upgrade patches differ from the Solaris patches.

The Release 5 Web Proxy Server upgrade patches for Linux OS are shown in the following table:

Table 13-6 Patches[1] to Upgrade Web Proxy Server on Linux

Description	Patch ID and RPM names
Web Proxy Server core	120983-10 • `sun-proxyserver-4.0-6.5.i386.rpm`
Web Proxy Server localization	122965-01 • `sun-proxyserver-`*Locale*`-4.0.4.i386.rpm`

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. 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 Web Proxy Server on the computer where the upgrade is taking place.

CAUTION An upgrade from Java ES Release 4 to Java ES Release 5 on Linux cannot be rolled back.

1. Log in as root or become superuser.

   ```
   su -
   ```

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

   WebProxyServer-base`/proxy-`*instanceName*`/stop`
   WebProxyServer-base`/proxy-admserv/stop`

3. If you have not already done so, synchronize all shared component to Release 5.

   See "Upgrade Web Proxy Server Dependencies" on page 251.

4. Obtain the required patches using the patch numbers and RPM names from Table 13-6. 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`

5. Apply the core and, if needed, localization RPMs for Web Proxy Server in Table 13-6, in that order.

   ```
   rpm -Fvh sun-proxyserver-version.i386.rpm
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -qa | grep sun-proxyserver
   ```

   The new version numbers of the RPMs should be returned.

7. Restart the Web Proxy Server instances that were stopped in Step 2.

   ```
   WebProxyServer-base/proxy-instanceName/start
   ```

Verifying the Upgrade

You can verify the upgrade of Web Proxy Server to Release 5 by starting a Web Proxy Server instance with the -version option:

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

See Table 13-4 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 253.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Web Proxy Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to Release 5 of Web Proxy Server is pretty much the reverse of the procedure for upgrading to Release 5.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

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

   ```
   WebProxyServer-base/proxy-instanceName/stop
   WebProxyServer-base/proxy-admserv/stop
   ```

3. Remove the patches in Table 13-5 on page 251.

   ```
   patchrm patch_ID
   ```

4. Restart the Web Proxy Server instances that were stopped in Step 2.

   ```
   WebProxyServer-base/proxy-instanceName/start
   ```

Upgrading Web Proxy Server from Version 3.6

This section includes information about upgrading Web Proxy Server from Version 3.6 to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Version 3.6 Web Proxy Server Upgrade

Introduction

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

- **General Upgrade Approach.** The upgrade is performed by using the Java ES installer to install Release 5 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 5.
- **Upgrade Dependencies.** Installation of shared components is automatically performed by the Java ES installer when upgrading Web Proxy Server to Release 5.
- **Backward Compatibility.** Release 5 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 5.
- **Upgrade Rollback.** Rollback of the Release 5 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.

Version 3.6 Web Proxy Server Upgrade

This section provides an overview of how to perform an upgrade of Web Proxy Server to Java ES Release 5. Web Proxy Server was not previously supported on the Linux platform. Hence upgrade of Web Proxy Server to Java ES Release 5 is only performed on the Solaris platform. The section covers the following topics:

- Pre-Upgrade Tasks
- Upgrading Version 3.6 Web Proxy Server
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Version 3.6 Upgrade

Pre-Upgrade Tasks

Before you upgrade Web Proxy Server, you should perform the following tasks:

- Verify Current Version Information
- Upgrade Web Proxy Server Dependencies
- Back Up Web Proxy Server Data
- Obtain Required Configuration Information and Passwords

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 13-7 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`
Release 5	`4.0.4`

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 5. However, the Java ES installer that is used to upgrade Web Proxy Server to Release 5 automatically upgrades all shared components upon which Web Proxy Server depends (see Table 1-9 on page 47).

Back Up Web Proxy Server Data

The Web Proxy Server upgrade to Release 5 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 Version 3.6 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 can be upgraded. However, the migration of configuration data has to be done separately for each instance. All such instances should be shut down when migration is performed to make sure that no port conflicts arise when migrated instance is started.

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 5.

   a. Run the Java ES installer from the Java ES Release 5 distribution.

   b. Select Web Proxy Server from the component selection page.

   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 4.0.4 Installation and Migration Guide,* `http://docs.sun.com/doc/819-5492`.

4. Make any additional configuration changes specified in Step f.

   Refer to the *Sun Java System Web Proxy Server 4.0.4 Configuration File Reference*, `http://docs.sun.com/doc/819-5494`, for more information.

Verifying the Upgrade

You can verify the upgrade of Web Proxy Server to Release 5 by starting a Web Proxy Server instance with the `-version` option:

WebProxyServer-base/*proxy-serverid*`/start -version`

See Table 13-7 on page 257 for output values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure” on page 258.

Rolling Back the Version 3.6 Upgrade

The upgrade of Web Proxy Server to Release 5, documented in "Upgrading Version 3.6 Web Proxy Server" on page 258, cannot be rolled back. However, you can revert to Version 3.6, which was left intact by the Release 5 upgrade procedure.

Chapter 14

Access Manager

This chapter describes how to upgrade Access Manager software from previous Java ES versions to Java ES 5 (Release 5): Sun Java System Access Manager 7.1.

The chapter provides a general overview of Access Manager upgrade issues and procedures for the different upgrade paths supported by Release 5. 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 values of these directory paths are shown in the following table.

Table 14-1 Access Manager Directory Paths

Path Name Variable	Solaris OS	Linux OS
AccessManager-base	`/opt/SUNWam`	`/opt/sun/identity`
AccessManagerConfig-base	`/etc/opt/SUNWam`	`/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 5 (Release 5):

- About Java ES Release 5 Access Manager
- Access Manager Upgrade Roadmap
- Access Manager Data
- Access Manager Upgrade Strategy

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 5 Access Manager

Java ES Release 5 Access Manager represents a minor release. It contains a number of bug fixes and enhancements to Java ES Release 4 Access Manager, which was a major release. Among the enhancements in Release 5 is a new monitoring capability based on the Java ES monitoring framework. For more information about Release 5 enhancements, see the *Sun Java System Access Manager 7.1 Release Notes*, `http://docs.sun.com/doc/819-4683.`

Similar to Release 4, Release 5 Access Manager supports multiple identity repositories, or user data stores. Thus Release 5 Access Manager supports not only an LDAP directory such as Directory Server, but other data storage protocols and formats as well.

On the front end, Access Manager Console is used to configure the new Access Manager services and identity repositories.

In order to provide backward compatibility with other Java ES components, Release 5 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 265).

Access Manager Upgrade Roadmap

Table 14-2 shows the supported Access Manager upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 14-2 Upgrade Paths to Java ES 5 (Release 5): Access Manager 7.1

Java ES Release	Access Manager Version	General Approach	Reconfiguration Required
Release 4	Sun Java System Access Manager 7.0 2005Q4	Direct upgrade: Performed by using a pre-upgrade script to remove the Release 4 version and then doing a full installation and reconfiguration of Release 5.	Configuration data Customized JSPs for Access Manager Console and authentication UI Directory schema
Release 3	Sun Java System Access Manager 6.3 2005Q1	Direct upgrade: Performed by using a pre-upgrade script to remove the Release 3 version and then doing a full installation and reconfiguration of Release 5.	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 using a pre-upgrade script to remove the Release 2 version and then doing a full installation and reconfiguration of Release 5.	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 5.	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 14-3 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 access control and configuration files	Web Server 7.0 (Java ES Release 5) `server.policy` and `server.xml` files in *WebServer7Config-base*/`https-`*configName*/`config` Web Server 6.*x* (Java ES Release 2, 3, and 4) `server.policy` and `server.xml` files in *WebServer6-base*/`https-`*hostname*/`config` Application Server 8.*x* (Java ES Release 3, 4, and 5): `server.policy` and `domain.xml` files in *AppServer8Config-base*/`domains/`*domainName*/`config` Application Server 7.*x* (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: (Java ES Release 2 and 3): *AccessManager-base*/`web-src/applications` Admin Console: (Java ES Release 4 and 5): *AccessManager-base*/`web-src/services` 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.

Table 14-3 Access Manager Data Usage

Type of Data	Location	Usage
Dynamic application data	None	Access Manager does not persistently store application data such as session state.

Access Manager Upgrade Strategy

Your strategy for upgrading Access Manager generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Access Manager by presenting issues that might influence your Access Manager upgrade plan.

Compatibility Issues

Release 5 Access Manager is backwardly compatible with Release 4 Access Manager, however Release 4 Access Manager was a major release that, except when configured to run in Legacy mode, broke compatibility with earlier releases. Similarly, Release 5 Access Manager, unless configured to run in Legacy mode, is not backwardly compatible with Release 3 Access Manager (or Release 4 Access Manager running in Legacy mode).

In addition, Release 5 Access Manager is not backwardly compatible with Release 2 Access Manager in any mode; Release 5 Access Manager cannot interoperate with Release 2 Access Manager SDK, nor *visa versa*.

Release 5 Access Manager, when configured to run in the newer Realm mode, supports multiple identity repositories and data storage protocols. Directory data has to be migrated to a new structure to support Realm mode operation. In addition, Realm mode does not support other Java ES components, such as Portal Server, or Sun Java Communications Suite components, such as Communications Express, Messaging Server, and others.

When configured to run in Legacy mode, however, Release 5 Access Manager, with some minor exceptions (see the *Sun Java System Access Manager 7.1 Release Notes*, `http://docs.sun.com/doc/819-4683)`, is backwardly compatible with Release 3 Access Manager and corresponding directory data.

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 5 Legacy mode.

Even when configured to run in Legacy mode, however, Release 5 Access Manager is not compatible with Release 3 or earlier Sun Java Communications Suite components. If Access Manager is upgraded to Release 5, then Release 3 or earlier Delegated Administrator also must be upgraded to Release 5 to provision users for Messaging Server and Calendar Server. However, Messaging Server and Calendar Server do not, themselves, have to be upgraded to Release 5.

Release 5 Access Manager Console, like the Release 4 Console, supports both Realm mode and Legacy mode. However, if you have configured Access Manager to run in Legacy mode, you can still use the Legacy-only Console that was distributed in Release 2 and Release 3.

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 1-9 on page 47). Access Manager upgrades might depend upon upgraded versions of these shared components.
- **Web Container.** Access Manager has a mandatory dependency on 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 might require that customized JSPs for the Access Manager Console or for the authentication UI be migrated to the upgraded Access Manager environment.
- **Directory Server.** Access Manager has a mandatory dependency on Directory Server, which is used to store configuration data and user data. As a result, Access Manager upgrades might require extensions of directory schema.

Web Container Upgrade Scenarios

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

Table 14-4 Web Container Upgrade Scenarios for Access Manager Upgrade

Scenario	Web Container in which Access Manager is Originally Deployed	Web Container in which Access Manager is Deployed After Upgrade	Applicable Access Manager Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 6.*x*	Release 2 Release 3 Release 4
Scenario 2	Web Server 6.*x*	Web Server 7.0	Release 2 Release 3 Release 4
Scenario 3	Application Server 8.1	Application Server 8.1	Release 3 Release 4
Scenario 4	Application Server 8.1	Application Server 8.2	Release 3 Release 4
Scenario 5	Application Server 7*x*	Application Server 8.2	Release 2

You must be careful when upgrading Access Manager (for example when using the `amconfig` script) to provide values appropriate to the upgrade scenario in Table 14-4 that applies, especially when there is a major version upgrade of the web container.

Dual Upgrade

Dual upgrades, in which both Access Manager and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) is not supported for Access Manager.

As a result, if you have a situation in which a dual upgrade is required, you have to perform an operating system install or upgrade, after which you re-install and freshly configure Access Manager.

Upgrading Access Manager from Java ES Release 4

This section includes information about upgrading Access Manager from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5). The section covers the following topics:

- Introduction
- Full Release 4 Access Manager Upgrade
- Multiple Instance Upgrades
- Release 4 Access Manager SDK-only Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is performed by removing previous versions of binaries and newly installing Release 5. An `ampre71upgrade` script is provided for removing the Release 4 version and the Java ES installer is then used to install Release 5. Reconfiguration of Access Manager is subsequently performed using the `amconfig` script, and directory schema is migrated using the `amupgrade` script.
- **Upgrade Dependencies.** Access Manager has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), all of which are automatically upgraded to Release 5 by the Java ES installer when you perform an upgrade of Access Manager. This includes support for the new Java ES monitoring framework, which requires a number of shared components not required for Release 4 Access Manager.

  In addition, Release 5 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 266. However, these are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Access Manager to Release 5.
- **Backward Compatibility.** Release 5 Access Manager is compatible with Release 4, but is not compatible with earlier Access Manager releases (see "Compatibility Issues" on page 265).

- **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. The best approach to rollback is to create a parallel installation using backed-up configuration files, and testing this parallel installation before performing the upgrade. This allows you to revert to the parallel installation if necessary.
- **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 4 Access Manager Upgrade

This section describes how to perform a full Access Manager upgrade from Java ES Release 4 to Java ES Release 5:

- Pre-Upgrade Tasks
- Upgrading Release 4 Access Manager
- Verifying the Access Manager Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade

Pre-Upgrade Tasks

Before you upgrade Access Manager, you should perform the following tasks:

- Verify Current Version Information
- Upgrade Access Manager Dependencies
- Back Up Directory Server Data
- Back Up Release 4 Access Manager Configuration Information
- Back Up Web Container Customized Files
- Back Up Release 4 Access Manager Log and Debug Files
- Back Up Custom Localization Files
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Access Manager using the following command:

```
AccessManager-base/bin/amadmin --version
```

Table 14-5 Access Manager Version Verification Outputs

Java ES Release	Access Manager Version Number
Release 2	`6.2`
Release 3	`6 2005Q1`
Release 4	`7 2005Q4`
Release 5	`7.1`

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 5. Access Manager has hard upgrade dependencies on a number of shared components (see Table 1-9 on page 47).

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

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63. However, all shared components required by Access Manager are upgraded automatically by the Java ES installer when you perform an upgrade of Access Manager to Release 5.

2. **Directory Server (soft upgrade dependency).** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, "Directory Server" on page 99.

3. **Web Container Software (soft upgrade dependency).** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, "Web Server" on page 133 and Chapter 11, "Application Server" on page 205, 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. You can use db2ldif to back up Access Manager schema and directory information tree (DIT).

For more information about backing up Directory Server, see the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*, http://docs.sun.com/doc/819-0995.

Back Up Release 4 Access Manager Configuration Information

Because the reconfiguration of Release 5 Access Manager software requires the reconfiguration of the Release 4 version, 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: see the location of the server.policy and server.xml files in Table 14-3 on page 264
  - For Application Server: see the location of the server.policy and domain.xml files in Table 14-3 on page 264
  - 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:
    - Realm/Legacy Console (distributed with Java ES Release 4)
      AccessManager-base`/web-src/services/`
    - Legacy-only Console (distributed with Java ES Release 2 and 3)
      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 4 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`

Back Up Custom Localization Files

If you have made any customization to the localized files installed by the Java ES installer or have added a new language localization that is not installed by the Java ES installer, then you should back up these customizations. The customizations might include the following:

- Customized Access Manager user interface localization
  AccessManager-base`/locale/*_`*Locale*`.properties`
- Customized authentication UI JSP pages
  AccessManager-base`/web-src/services/config/auth/default_`*Locale*
- Customized online help translations
  AccessManager-base`/web-src/services/html/`*Locale*

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 4 Access Manager

The upgrade of Access Manager software to Java ES Release 5 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. Upgrade Access Manager mobile access software.
2. Remove the Java ES Release 4 Version of Access Manager. Use the `ampre71upgrade` script.
3. If the upgrade to Release 5 needs to be localized, remove the Release 4 localization packages. This step has to be performed by hand.
4. Install the Java ES Release 5 Version of Access Manager. Use the Java ES installer with the Configure Later option.
5. Re-customize JSPs for Access Manager.
6. Undeploy Access Manager, re-configure, and re-deploy into a Web Container. Use the `amconfig` script.
7. Update the directory structure and schema. Use the `amupgrade` script.

These steps are each documented in the following procedures.

Upgrade Procedure

1. Upgrade Access Manager mobile access software.

   Access Manager mobile access software needs to be upgraded by patching the Release 4 version. The patches needed are shown in the following table:

Table 14-6 Patches[1] to Upgrade Access Manager Mobile Access Software

Description	Patch ID: Solaris 9 & 10	Patch ID: Linux
Mobile Access software	119530-05 (SPARC) 119531-05 (x86)	119532-05 • `sun-identity-mobileaccess-6.2-25.3.i386.rpm` • `sun-identity-mobileaccess-config-6.2-25.3.i386.rpm`

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. 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 14-6.

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

b. Perform any pre-patch procedures indicated in the patch `README` files.

c. Obtain the values of the following parameters to be requested by the patch:

Table 14-7 Mobile Access Patch Parameters

Parameter	Value
Directory Manager’s DN	default: `cn=Directory Manager`
Directory Manager’s Password	

d. Apply the patches in Table 14-6.

   On Solaris:
   `patchadd /tmp/`*`patch_ID`*

   On Linux:
   `./update`

Perform any post-patch procedures indicated in the patch `README` files.

2. Remove the Java ES Release 4 Version of Access Manager.

   a. Log in as root to the computer hosting Release 4 Access Manager or become superuser.

```
su -
```

b. Change directory to the *os_arch*/Product/identity_svr/Tools directory in the Java ES Release 5 distribution, where *os_arch* matches your platform, such as Solaris_sparc.

c. Obtain the values of the following parameters to be requested by the ampre71upgrade script:

Table 14-8 Access Manager Configuration Parameters: ampre71upgrade

Parameter	Value
Directory Server Host	Set the fully-qualified name: *hostname.domain*
Directory Server Port	Specify a non-SSL port number[1] Default: 389
Top-Level Administrator DN	Default: uid=amadmin,ou=People,*default_org_DN*
Top-Level Administrator Password	
Directory to store back up files	Default: *AccessManager-base*

[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 ampre71upgrade script.

```
./ampre71upgrade
```

The script backs up Access Manager configuration files and removes Release 4 base packages (localized packages must be removed manually per Step 3, below).

3. If the upgrade to Release 5 needs to be localized, remove the Release 4 localization packages.

The ampre71upgrade script run in Step 2 above does not remove localization packages, so you have to remove them manually, as follows.

On Solaris:

a. Check for localization packages.

```
pkginfo | grep SUNWaml
pkginfo | grep SUNWamclnt
pkginfo | grep SUNWamdistauth
```

b. Remove any localization packages found in Step a above.

```
pkgrm SUNWamlLocale
pkgrm SUNWamclntLocale
pkgrm SUNWamdistauthLocale
```

On Linux:

a. Check for localization RPMs.

```
rpm -qa | grep sun-identity-sdk-*
rpm -qa | grep sun-identity-clientsdk-*
rpm -qa | grep sun-identity-distauth-*
```

b. Remove any localization RPMs found in Step a above.

```
rpm -e sun-identity-sdk-Locale-*
rpm -e sun-identity-clientsdk-Locale-*
rpm -e sun-identity-distauth-Locale-*
```

4. Install the Java ES Release 5 Version of Access Manager.

Perform the following steps:

a. Launch the Java ES installer on the computer hosting Release 4 Access Manager.

```
cd Java ES Release 5 distribution/os_arch
./installer
```

where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of “upgradable.”)

b. Select Access Manager from the component selection page.

c. Specify the same installation directory in which Release 4 was installed.

d. Select the Configure Later option.

e. If needed, select the option to install localized packages.

f. Exit the Java ES installer when installation is complete.

5. Re-customize JSPs for Access Manager.

Re-apply the Release 4 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 272.

Then, copy the customized JSP files to the correct directories:

- Realm/Legacy Access Manager Console
  AccessManager-base`/web-src/services/console`
- Legacy-only Access Manager 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 7.1 Developer's Guide*, `http://docs.sun.com/doc/819-4675`.

6. 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 7.1 Administration Guide*, `http://docs.sun.com/doc/819-4670`.

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 270, make sure the upgrade is complete.

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

Table 14-9 Administrative Server Modes Supported by `amconfig`

Web Container	Supported Mode	Default Port Number
Application Server (8.*x*): Java ES Release 3, 4, & 5	SSL (secure) non-SSL	`4849`
Web Server (7.0): Java ES Release 5	SSL (secure)	`8989`
Web Server (6.*x*): Java ES Release 2. 3, & 4	non-SSL	`8888`

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

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

These would likely be components such as Portal Server or Sun Java Communications Suite Components such as Communications Express, Instant Messaging, or Delegated Administrator.

The procedure is as follows:

I. Log in as admin at `https://host:8989`

II. Go to Edit Virtual Server.

III. Select the Web Applications tab.

IV. Check all Access Manager dependent applications.

V. Click Disable.

VI. Click Save.

VII. Click deployment pending | Deploy Config.

The configuration change will propagate to the Web Server instance.

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

f. Create an `amconfig` input file based on the `amsamplesilent` template input file:

`cp amsamplesilent` *config-file*

(In subsequent steps, *config-file* is assumed to reside in the same directory as `amsamplesilent`.)

g. 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 14-10 Access Manager Configuration Parameters: `amconfig`

Parameter	Value
Upgrade Parameters	
DEPLOY_LEVEL	Set to `26` for undeploy or Set to `1` for re-configure and deploy
DIRECTORY_MODE	Set to `5`
AM_REALM[1]	Set to `disabled` if Legacy Mode is enabled) Set to `enabled` if Realm Mode is enabled Default: `enabled`
JAVA_HOME	Set to JDK Release 5 directory: `/usr/java/jdk1.5.0_04/`
WEB_CONTAINER	Set to `WS` for Web Server 7.*x* Set to `WS6` for Web Server 6.*x* Set to `AS8` for Application Server 8.*x* Set to `WAS5` for IBM WebSphere 5.*x* Set to `WL8` for BEA WebLogic 8.*x* and fill out only the corresponding section of *config-file*.
WS_INSTANCE (If using Web Server 7.*x* as the web container)	Set to the case-sensitive instance configuration directory name: `https-`*configName/* The directory is in the following path: *WebServer7Config-base/*`https-`*configName/*
WS61_INSTANCE (If using Web Server 6.*x* as the web container	Set to the case-sensitive instance configuration directory name: `https-`*instanceName* The directory is in the following path: *WebServer6-base/*`https-`*instanceName/*
AS81_INSTANCE (Using Application Server 8.*x* as the web container)	Set to Application Server 8.*x* *instanceName* Default: `server`

Table 14-10 Access Manager Configuration Parameters: `amconfig` *(Continued)*

Parameter	Value
AS81_INSTANCE_DIR (Using Application Server 8.*x* as the web container)	Set to the Application Server 8.*x* domain directory for the instance, which, by default is *AppServer8Config-base*`/domains/domain1`
AS81_DOCS_DIR (Using Application Server 8.*x* as the web container)	Set to the Application Server 8.*x* docroot directory for the instance, which, by default is *AppServer8Config-base*`/domains/domain1/docroot`
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[2]	`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[2]	`am.encryption.pwd`[3]

[1] For more information on Realm and Legacy modes, see "Compatibility Issues" on page 265.

[2] The value of this parameter should be the same as in the previous version of Access Manager.

[3] When Access Manager and Access Manager SDK are both deployed, the value of this property must be the same for both the Access Manager instance and its associated Access Manager SDK instance.

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

Table 14-11 `amconfig` Parameters: Release 5 Web Server

Parameter	Value
WS_CONFIG	The name of the Web Server configuration: *configName*
WS_INSTANCE	`https-`*configName*
WS_HOME	*WebServer7Config-base*
WS_PROTOCOL	`http` or `https`
WS_HOST	Fully qualified hostname on which Web Server instance is listening for connections
WS_PORT	Port on which Web Server instance is listening for connections
WS_ADMINPORT	Port on which Web Server administration instance is listening for connections
WS_ADMIN	Web Server administrator User ID
WS_ADMINPASSWD	Web Server administrator password

h. 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
```

i. Check to make sure that the Common Agent Container is running.

```
netstat -an | grep 11163
```

If it is not running, start it up.

```
/usr/sbin/cacaoadm start
```

j. 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. Update the directory structure and schema.

Release 5 Access Manager co-exists with the Release 4 directory structure, but the structure must be modified to support Release 5 capabilities. Update the Access Manager directory structure and schema to Release 5 by running the `amupgrade` script, which is installed in the following directory:

- *On Solaris:*
AccessManager-base`/upgrade/scripts`
- *On Linux:*
AccessManager_base`/identity/upgrade/scripts`

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

Table 14-12 Access Manager Configuration Parameters: `amupgrade`

Parameter	Value
Directory Server fully qualified host name	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,`*default_org_DN*
Top-Level Administrator Password	
Enable Realm Mode (This parameter value not requested when upgrading from Release 4 Realm Mode.)	`Y/N`: `Yes` means Realm Mode is enabled and services data is migrated to new Realm tree[2]. `No` (default) means services data remain in Legacy Mode.

[1] You must specify a Directory Server SSL port different from the default SSL value of 636.

[2] See "Migrating to Realm Mode" on page 284.

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:

   On Solaris:

   ```
   /var/sadm/install/logs/
          Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
   ```

   On Linux:

   ```
   /var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
   ```

8. Enable any components that were disabled in Step d on page 278.

9. Re-start the web container in which Access Manager is deployed.

10. 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. Check the upgrade of Access Manager packages using the following command:

   AccessManager-base`/bin/amadmin --version`

   See Table 14-5 on page 270 for output values.

2. Review the status of the upgrade by checking the following installer log files in the `/var/sadm/install/logs` directory:

   - `Java_Shared_Component_Install.`*timestamp*
   - `Java_Enterprise_System_install.`*Atimestamp*
   - `Java_Enterprise_System_install.`*Btimestamp*
   - `Java_Enterprise_System_Summary_Report_install.`*timestamp*

3. Review the status of the Access Manager migration by checking the terminal window for errors while running the `amupgrade` script.

   Also, check the following log file in the `/var/sadm/install/logs` directory:

   `Sun_Java_System_Access_Manager_upgrade_dit_log.`*timestamp*

4. Review Access Manager trouble shooting files for errors.

   The files are located at the location specified in the com.iplanet.services.debug.directory property of the AMConfig.properties file. The default values are:

   On Solaris:
   `/var/opt/SUNWam/debug`

   On Linux:
   `/var/opt/sun/identity/debug`

Post-Upgrade Tasks

Please note the post-upgrade procedures required to address the following situations:

- Migrating to Realm Mode
- Security Assertion Markup Language

Migrating to Realm Mode

If you have migrated to Realm Mode when upgrading Access Manager to Release 5, (that is, when using `amupgrade` to update the directory structure and schema, you answer `Yes` to enabling Realm mode), then perform the following steps:

1. Open the *AccessManagerConfig-base/*`config/AMConfig.properties` file.
2. Check the value of the following property:

   ```
   com.sun.identity.sm.ldap.enableProxy
   ```

3. If the property is not set to `false`, then manually set it to `false`.

Security Assertion Markup Language

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 7.1 Administration Guide,* `http://docs.sun.com/doc/819-4670`.

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 4 Access Manager Log and Debug Files" on page 272). Backing out the upgrade is too difficult to be practical.

One approach to rollback is to perform a re-install of Release 4 and migrate all the backed-up configuration files to their rightful locations. Another is to create a parallel system before upgrading, using the backed-up configuration files, and testing the parallel system before attempting an upgrade.

Multiple Instance Upgrades

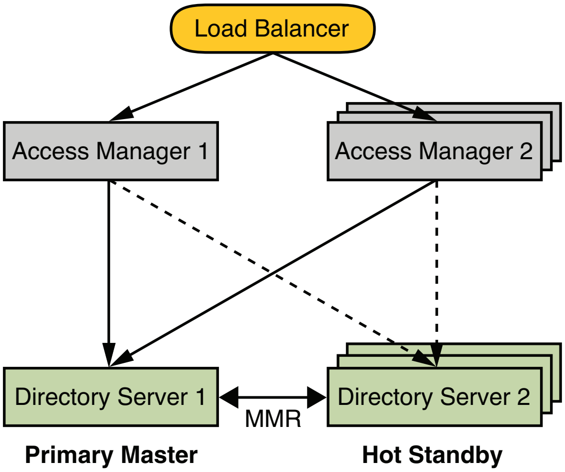
In some deployment architectures Access Manager is deployed on multiple computer systems to provide for high availability and scalability.

It is usually desirable to upgrade the Access Manager instances sequentially without interrupting service. This section discusses the procedure for performing such rolling upgrades from Release 4 Access Manager to Release 5.

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 deployment architecture shown in the following figure will be used to illustrate the rolling upgrade procedure.

Figure 14-1 Example Deployment Architecture for Multiple Access Manager Instances



In this architecture, multiple Access Manager instances are accessed through a load balancer, and these instances, in turn, access a directory that is set up for multi-master replication (MMR). While other Directory Server replication schemes are possible, MMR is representative of highly available and scalable directory services. In Figure 14-1, the multiple instances of Access Manager and Directory Server are grouped to facilitate explanation of the upgrade procedure. Access Manager 2, for example, is representative of the second through nth instances of Access Manager.

The procedure for performing a rolling upgrade from Release 4 Access Manager to Release 5 is based on the following interoperability: Release 5 Access Manager and Release 4 Access Manager instances can coexist and run concurrently against the same directory if the directory schema has not yet been updated to Release 5.

Hence, for Access Manager instances that point to a single Directory Server instance, you can perform a rolling upgrade by delaying the update of the directory schema until all Access Manager instances have been upgraded.

You can perform a rolling upgrade from Release 4 Access Manager to Release 5 using the following procedure:

1. Back up Release 4 configuration information on all Access Manager instances.

   See Table 14-3 on page 264.

2. Upgrade Access Manager 1.

   a. Disable Access Manager 1 in the load balancer.

      Requests will no longer be routed to Access Manager 1.

   b. Partially upgrade Access Manager 1.

      Upgrade Access Manager as described in "Upgrading Release 4 Access Manager" on page 273, except for updating the directory structure and schema, Step 7 on page 282.

   c. Enable Access Manager 1 in the load balancer.

3. Upgrade Access Manager 2 through Access Manager n.

   For brevity, in succeeding steps, "Access Manager 2" will mean Access Manager 2 through Access Manager n.

   a. Disable Access Manager 2 in the load balancer.

      Requests will no longer be routed to Access Manager 2.

   b. Partially upgrade Access Manager 2.

      Upgrade each instance of Access Manager as described in "Upgrading Release 4 Access Manager" on page 273, except for updating the directory structure and schema, Step 7 on page 282.

   c. Enable Access Manager 2 in the load balancer.

      Requests will be once again routed to Access Manager 2.

4. Update the directory structure and schema for Directory Server 1.

   Use the amupgrade script as documented in Step 7 on page 282. Access Manager 1 through n will continue to function when the schema for Directory Server 1 has been updated.

Release 4 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.

Access Manager SDK and the full Access Manager for which it serves as a remote interface should both be upgraded to Release 5. However Release 5 Access Manager is backwardly compatible with Release 4 Access Manager SDK, so Access Manager should generally be upgraded first before upgrading Access Manager SDK on other computers.

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, 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 4 to Java ES Release 5:

- Pre-Upgrade Tasks
- Upgrading Release 4 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 (see “Pre-Upgrade Tasks” on page 269), but exclude those tasks related to Directory Server and to Access Manager administration tool JSP customizations. The pre-upgrade tasks needed for Access Manager SDK are the following:

- “Upgrade Access Manager Dependencies” on page 270

  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 4 Access Manager Configuration Information” on page 271

  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 4 Access Manager Log and Debug Files" on page 272

  You also need to obtain the admin username and password for accessing these files.

Upgrading Release 4 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 localization, Access Manager administration tool JSP customizations, and migrating directory schema.

1. Remove the Java ES Release 4 version of Access Manager SDK.

   Follow the instructions in "Remove the Java ES Release 4 Version of Access Manager." on page 274, except remove only Access Manager SDK.

2. Install Java ES Release 5 version of Access Manager SDK.

   Follow the instructions in "Install the Java ES Release 5 Version of Access Manager." on page 276, 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 277, except set 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 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 4 Access Manager Log and Debug Files” on page 272). Backing out the upgrade is too difficult to be practical.

One approach to rollback is to perform a re-install of R4 and migrate all the backed-up configuration files to their rightful locations. Another is to create a parallel system before upgrading, using the backed-up configuration files, and testing the parallel system before attempting an upgrade.

Upgrading Access Manager from Java ES Release 3

The procedure for upgrading Java ES 2003Q1 (Release 3) Access Manager or Access Manager SDK to Release 5 is similar to that for upgrading Release 4 Access Manager or Access Manager SDK to Release 5, with the exception of how to perform multi-instance upgrades.

- Release 3 Access Manager Upgrade
- Multiple Instance Upgrades

NOTE Release 5 Access Manager is not compatible with some Release 3 Sun Java Communications Suite components. If Access Manager is upgraded to Release 5, then Release 3 or earlier Delegated Administrator also must be upgraded to Release 5 to provision users for Messaging Server and Calendar Server. However, Messaging Server and Calendar Server do not, themselves, have to be upgraded to Release 5.

Release 3 Access Manager Upgrade

To upgrade Release 3 Access Manager or Access Manager SDK to Release 5, use the instructions in "Upgrading Access Manager from Java ES Release 4" on page 268, except substitute Release 3 wherever Release 4 is referenced.

Multiple Instance Upgrades

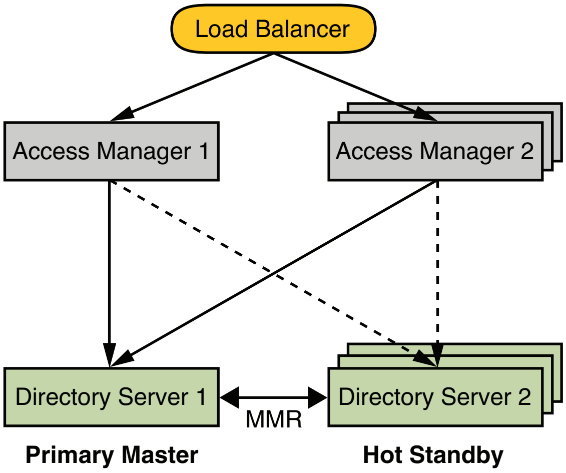
In some deployment architectures Access Manager is deployed on multiple computer systems to provide for high availability and scalability.

It is usually desirable to upgrade multiple Access Manager instances sequentially without interrupting service. This section discusses the procedure for performing such rolling upgrades from Release 3 Access Manager to Release 5.

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 deployment architecture shown in the following figure will be used to illustrate the rolling upgrade procedure.

Figure 14-2 Example Deployment Architecture for Multiple Access Manager Instances



In this architecture, multiple Access Manager instances are accessed through a load balancer, and these instances, in turn, access a directory that is set up for multi-master replication (MMR). While other Directory Server replication schemes are possible, MMR is representative of highly available and scalable directory services. In Figure 14-2, the multiple instances of Access Manager and Directory Server are grouped to facilitate explanation of the upgrade procedure. Access Manager 2, for example, is representative of the second through nth instances of Access Manager.

The procedure for performing a rolling upgrade of Release 3 Access Manager to Release 5 is based on the following constraint: Release 5 Access Manager can *not* co-exist with the Release 3 directory structure. However, if Directory Server instances are replicated, as in Figure 14-2, then you can perform a rolling upgrade using the following procedure:

1. **Back up Release 3 configuration information on all Access Manager instances.**

   See Table 14-3 on page 264.

2. **Modify the configuration of `Access Manager 1`.**

   a. **Configure `Access Manager 1` to point to `Directory Server 2` rather than `Directory Server 1`.**

   b. **Restart `Access Manager 1`.**

   `Access Manager 1` will continue handling requests while `Access Manager 2` through `Access Manager n` will be upgraded.

3. **Upgrade `Access Manager 2` through `Access Manager n`.**

   For brevity, in succeeding steps, "`Access Manager 2`" will mean `Access Manager 2` through `Access Manager n`.

   a. **Disable `Access Manager 2` in the load balancer.**

      Requests will no longer be routed to `Access Manager 2`.

   b. **Partially upgrade `Access Manager 2`.**

      Upgrade each instance of Access Manager as described in "Upgrading Release 4 Access Manager" on page 273, except for updating the directory structure and schema, Step 7 on page 282.

   c. **Disable Directory Server MMR.**

   d. **Update the directory structure and schema for `Directory Server 1`.**

      Use the `amupgrade` script as documented in Step 7 on page 282. `Access Manager 1` will continue to function because the schema for `Directory Server 2` is not being updated.

   e. **Restart `Access Manager 2`.**

   f. **Enable `Access Manager 2` in the load balancer.**

      Requests will be once again routed to `Access Manager 2`.

4. **Upgrade** `Access Manager 1`.

   a. Disable `Access Manager 1` in the load balancer.

      Requests will no longer be routed to `Access Manager 1`.

   b. Partially upgrade `Access Manager 1`.

      Upgrade Access Manager as described in "Upgrading Release 4 Access Manager" on page 273, except for updating the directory structure and schema, Step 7 on page 282.

   c. Enable Directory Server MMR.

      The schema (and data) for `Directory Server 2`, is now updated.

   d. Restore the configuration of `Access Manager 1` to point to `Directory Server 1`.

   e. Restart `Access Manager 1`.

   f. Enable `Access Manager 1` in the load balancer.

   Requests will be once again routed to `Access Manager 1` as well as to all other upgraded Access Manager instances.

Upgrading Access Manager from Java ES Release 2

The procedure for upgrading Java ES 2004Q2 (Release 2) Access Manager to Release 5 is similar to that for upgrading Release 4 Access Manager to Release 5, with a few differences, as indicated in the sections below:

- Pre-Upgrade Tasks
- Release 2 Access Manager Upgrade
- Multiple Instance Upgrades

Also, the procedure for upgrading Java ES 2004Q2 (Release 2) Access Manager SDK to Release 5 is similar to that for upgrading Release 4 Access Manager SDK to Release 5 (see "Release 4 Access Manager SDK-only Upgrades" on page 287), with similar exceptions. Access Manager SDK-only upgrade excludes procedures related to localization, Access Manager administration tool JSP customizations, and migrating directory schema.

Release 2 Access Manager SDK and the full Release 2 Access Manager for which it serves as a remote interface must both be upgraded to Release 5. Mixtures of Release 2 and Release 5 components are not supported. Hence, all instances of Release 2 Access Manager and Release 2 Access Manager SDK on all computers must be upgraded to Release 5.

NOTE If you are upgrading from Release 2 Access Manager on the Linux platform, then you will have to perform a dual upgrade, in which both Access Manager *and* the operating system are upgraded (Release 5 Access Manager is not supported on RHEL 2.1). See "Dual Upgrade" on page 267 for more information.

Pre-Upgrade Tasks

Before you upgrade Access Manager, perform the procedures described in "Pre-Upgrade Tasks" on page 269, with the following exceptions and additions:

- Upgrade Access Manager Dependencies
- Upgrade Directory Schema
- Re-index the Directory

Upgrade Access Manager Dependencies

As compared to the upgrade from Release 4, the Release 2 to Release 5 pre-upgrade tasks require the upgrading to Release 5 of all shared components (see Table 1-9 on page 47) 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 synchronizing Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63. However, Java ES shared components will be upgraded automatically by the installer when you perform a fresh install of Release 5 Access Manager.

2. **Directory Server.** Directory Server rarely resides on the same computer as Access Manager, however, instructions for upgrading Directory Server to Release 5 are provided in "Upgrading Directory Server from Java ES Release 2" on page 115.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in "Upgrading Web Server from Java ES Release 3" on page 156 and "Upgrading Application Server from Java ES Release 2" on page 220, respectively.

Upgrade Directory Schema

If Directory Server was configured with Sun Java Communications Suite's Directory Preparation Tool (`comm_dssetup.pl`) to support Communication Suite components, such as Messaging Server and Calendar Server, you must first upgrade the directory schema using Directory Preparation Tool 6.4 *before* upgrading Access Manager (see the *Sun Java Communications Suite 5 Upgrade Guide*, `http://docs.sun.com/doc/819-7561`). Perform this pre-upgrade task after you have upgraded Access Manager dependencies.

Re-index the Directory

In order to avoid complications when performing the upgrade of Access Manager after having upgraded the directory schema (see "Upgrade Directory Schema," above), you need to manually re-index the Access Manager directory root suffix, as follows:

Release 2-Release 4 Directory Server:

1. `cd` *serverRoot*`/slapd-`hostname``

2. `./db2index.pl -D "cn=directory manager" -w` *passwordFile* `-n` *databaseName*

   where the default *databaseName* is `userRoot`.

Release 5 Directory Server:

1. `cd` *DirServer-base*`/ds6/bin`

2. `./dsconf reindex -D "cn=Directory Manager" -e -w` *passwordFile* *suffix*

   where

   `-e` signifies an unsecure connection

   `-D` is the Directory Manager

   `-w` is a password file containing just the password

   suffix is the Access Manager directory root suffix.

Depending on the number of entries in the directory, it can take a significant amount of time to complete the re-indexing.

Release 2 Access Manager Upgrade

The procedure for upgrading Access Manager from Release 2 to Release 5 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 5, when deploying into a Web Server web container, follow the instructions in "Upgrading Release 4 Access Manager" on page 273, except substitute Release 2 wherever Release 4 is referenced.

Upgrading Release 2 Access Manager: Application Server Web Container

To upgrade Release 2 Access Manager to Release 5, when deploying into an Application Server web container, there are two cases:

- **Release 5 Application Server has been freshly installed.** In this case, to upgrade Release 2 Access Manager to Release 5, follow the instructions in "Upgrading Release 4 Access Manager" on page 273, except substitute Release 2 wherever Release 4 is referenced.
- **Release 2 Application Server has been upgraded to Release 5.** In this case, the Release 2 Application Server instance in which Access Manager was originally deployed (*instanceName*), when upgraded to Release 5, was migrated under a node agent created by the upgrade process. Upgrade of Access Manager in this upgraded Application Server instance requires the steps in the following sections:

Upgrade Summary

The procedure for upgrading Access Manager consists of the following steps:

1. Upgrade Access Manager mobile access software.
2. Remove the Java ES Release 2 Version of Access Manager. Use the `ampre71upgrade` script.
3. If the upgrade to Release 5 needs to be localized, remove the Release 2 localization packages. This step has to be performed by hand.
4. Install the Java ES Release 5 Version of Access Manager. Use the Java ES installer with the Configure Later option.
5. Re-customize JSPs for Access Manager.
6. Check that Directory Server is running.
7. Start the following Application Server instances: Domain Administration Server (DAS), node agent, and server instance in which Access Manager is deployed.
8. Undeploy Access Manager, reconfigure, and re-deploy into the Application Server instance. Use the `amconfig` script
9. Verify that Access Manager `classpath-suffix` and `server-classpath` information have been migrated to the Release 5 Application Server `domain.xml` file.
10. Stop the Domain Administration Server (DAS), node agent, and server instance.
11. Restart the Domain Administration Server (DAS), node agent, and server instance.
12. Update the directory structure and schema. Use the `amupgrade` script.

These steps are documented in the following procedure.

Upgrade Procedure

1. Upgrade Access Manager mobile access software.

   Access Manager mobile access software needs to be upgraded by patching the Release 2 version. The patches needed are shown in the following table:

   Table 14-13 Patches[1] to Upgrade Access Manager Mobile Access Software

Description	Patch ID: Solaris 9 & 10	Patch ID: Linux
Mobile Access software	119530-05 (SPARC) 119531-05 (x86)	119532-05 • `sun-identity-mobileaccess-6.2-25.3.i386.rpm` • `sun-identity-mobileaccess-config-6.2-25.3.i386.rpm`

   [1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. 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 14-6.

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

   b. Perform any pre-patch procedures indicated in the patch `README` files.

   c. Obtain the values of the following parameters to be requested by the patch:

      Table 14-14 Mobile Access Patch Parameters

Parameter	Value
Directory Manager’s DN	default: `cn=Directory Manager`
Directory Manager’s Password	

d. Apply the patches in Table 14-6.

On Solaris:

```
patchadd patch_ID
```

On Linux:

```
./update
```

Perform any post-patch procedures indicated in the patch README files.

2. Remove the Java ES Release 2 Version of Access Manager.

a. Log in as root to the computer hosting Release 4 Access Manager or become superuser.

```
su -
```

b. Change directory to the *os_arch*/Product/identity_svr/Tools directory in the Java ES Release 5 distribution, where *os_arch* matches your platform, such as Solaris_sparc.

c. Obtain the values of the following parameters to be requested by the ampre71upgrade script:

Table 14-15 Access Manager Configuration Parameters: ampre71upgrade

Parameter	Value
Directory Server Host	Set the fully-qualified name: *hostname.domain*
Directory Server Port	Specify a non-SSL port number[1] Default: 389
Top-Level Administrator DN	Default: uid=amadmin,ou=People,*default_org_DN*
Top-Level Administrator Password	
Directory to store back up files	Default: *AccessManager-base*

[1] You must specify a Directory Server SSL port different from the default SSL value of 636.

d. Make sure that Directory Server is running or start it if it is not.

e. Run the ampre71upgrade script.

```
./ampre71upgrade
```

The script backs up Access Manager configuration files and removes Release 4 base packages (localized packages must be removed manually per Step 3, below).

3. If the upgrade to Release 5 needs to be localized, remove the Release 2 localization packages.

   The ampre71upgrade script run in Step 2 above does not remove localization packages, so you have to remove them manually, as follows.

   On Solaris:

   a. Check for localization packages.

   ```
   pkginfo | grep SUNWaml
   pkginfo | grep SUNWamclnt
   pkginfo | grep SUNWamdistauth
   ```

   b. Remove any localization packages found in Step a above.

   ```
   pkgrm SUNWamlLocale
   pkgrm SUNWamclntLocale
   pkgrm SUNWamdistauthLocale
   ```

   On Linux:

   a. Check for localization RPMs.

   ```
   rpm -qa | grep sun-identity-sdk-*
   rpm -qa | grep sun-identity-clientsdk-*
   rpm -qa | grep sun-identity-distauth-*
   ```

   b. Remove any localization RPMs found in Step a above.

   ```
   rpm -e sun-identity-sdk-Locale-*
   rpm -e sun-identity-clientsdk-Locale-*
   rpm -e sun-identity-distauth-Locale-*
   ```

4. Install the Java ES Release 5 Version of Access Manager.

   Perform the following steps:

   a. Launch the Java ES installer on the computer hosting Release 2 Access Manager.

   ```
   cd Java ES Release 5 distribution/os_arch
   ./installer
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`. (Use the `installer -nodisplay` option for the command line interface.)

After the Welcome and License Agreement pages are displayed, you will be presented with a component selection page. (When installed components are detected that can be directly upgraded by the Java ES installer, they are shown with a status of "upgradable.")

b. Select Access Manager from the component selection page.

c. Specify the same installation directory in which Release 2 was installed.

d. Select the Configure Later option.

e. If needed, select the option to install localized packages.

f. Exit the Java ES installer when installation is complete.

5. Re-customize JSPs for Access Manager.

   Re-apply the Release 2 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 272.

   Then, copy the customized JSP files to the correct directories:

   - Legacy-only Access Manager 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 7.1 Developer's Guide*, `http://docs.sun.com/doc/819-4675`.

6. Check that Directory Server is running.

7. Start the following Application Server instances:

   In the following commands, and in subsequent steps, the following conventions are used:

   - where *nodeagentName* has the form *hostName_domainName*, but is simply *hostName* by default

   - The default *domainName* is `domain1`

   - The default *instanceName* is `server1`

NOTE Be sure to separately start the node agent, as shown below, using the `startinstances=false` option before starting the server instance.

a. Start the Domain Administration Server (DAS)

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
    domainName
```

b. Start the node agent under which the upgraded Application Server instance has been migrated

```
AppServer8-base/bin/asadmin start-node-agent
--startinstances=false --user admin_ID  nodeagentName
```

c. Start the server instance in which Access Manager is deployed (*instanceName*), if that server instance is not already running.

```
AppServer8-base/bin/asadmin start-instance --user admin_ID
    instanceName
```

8. Undeploy Access Manager, reconfigure, and re-deploy into the Application Server instance.

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

b. Create an `amconfig` input file based on the `amsamplesilent` template input file:

```
cp amsamplesilent config-file
```

(In subsequent steps, *config-file* is assumed to reside in the same directory as `amsamplesilent`.)

c. 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 14-16 Access Manager Configuration Parameters: `amconfig`

Parameter	Value
Upgrade Parameters	
DEPLOY_LEVEL	Set to `26` for undeploy or Set to `1` for re-configure and deploy
DIRECTORY_MODE	Set to `5`
AM_REALM[1]	Set to `disabled` if Legacy Mode is enabled) Set to `enabled` if Realm Mode is enabled Default: `enabled`

Table 14-16 Access Manager Configuration Parameters: `amconfig` *(Continued)*

Parameter	Value
JAVA_HOME	Set to JDK Release 5 directory: `/usr/java/jdk1.5.0_04/`
WEB_CONTAINER	Set to `AS8` for Application Server 8.*x* and fill out only the corresponding section of *config-file*.
AS81_INSTANCE (Using Application Server 8.*x* upgraded from Application Server 7.*x* as the web container)	Set to Application Server 7.*x* *instanceName,* which, by default is `server1`
AS81_INSTANCE_DIR (Using Application Server 8.*x* as the web container)	Set to the Application Server 8.*x* domain directory for the instance, which, by default is *AppServer8Config-base*`/domains/domain1`
AS81_DOCS_DIR (Using Application Server 8.*x* as the web container)	Set to the Application Server 8.*x* docroot directory for the instance, which, by default is *AppServer8Config-base*`/domains/domain1/docroot`
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[2]	`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[2]	`am.encryption.pwd`[3]

[1] For more information on Realm and Legacy modes, see "Compatibility Issues" on page 265.

[2] The value of this parameter should be the same as in the previous version of Access Manager.

[3] When Access Manager and Access Manager SDK are both deployed, the value of this property must be the same for both the Access Manager instance and its associated Access Manager SDK instance.

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.

d. 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
```

e. Check to make sure that the Common Agent Container is running.

```
netstat -an | grep 11163
```

If it is not running, start it up.

```
/usr/sbin/cacaoadm start
```

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
```

9. Verify that Access Manager `classpath-suffix` and `server-classpath` information have been migrated to the Release 5 Application Server `domain.xml` file.

a. Note 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. Check that the `classpath-suffix` and `server-classpath` entries, have been appended to 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 5 Application Server `domain.xml` file. This block begins with the following line:

```
<config dynamic-reconfiguration-enabled="true"
name="instanceName-config">
```

10. Update the directory structure and schema.

    Release 5 Access Manager co-exists with the Release 4 directory structure, but the structure must be modified to support Release 5 capabilities. Update the Access Manager directory structure and schema to Release 5 by running the `amupgrade` script, which is installed in the following directory:

    - *On Solaris:*
      AccessManager-base`/upgrade/scripts`
    - *On Linux:*
      AccessManager_base`/identity/upgrade/scripts`

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

Table 14-17 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,`*default_org_DN*
Top-Level Administrator Password	
Enable Realm Mode (This parameter value not requested when upgrading from Release 4 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:

On Solaris:

```
/var/sadm/install/logs/
       Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

On Linux:

```
/var/log/Sun_Java_System_Access_Manager_upgrade_dit_log.mmddhhmm
```

11. Stop the Domain Administration Server (DAS), node agent, and server instance.

    These are the instances that were started in Step 7 on page 302.

    ```
    AppServer8-base/bin/asadmin stop-domain --user admin_ID
         domainName
    ```

    ```
    AppServer8-base/bin/asadmin stop-node-agent --user admin_ID
         nodeagentName
    ```

12. Restart the Domain Administration Server (DAS), node agent, and server instance.

    NOTE Be sure to separately start the node agent using the `startinstances=false` option before starting the server instance, as shown below.

    ```
    AppServer8-base/bin/asadmin start-domain --user admin_ID
         domainName
    ```

    ```
    AppServer8-base/bin/asadmin start-node-agent --port DASportNumber
         --startinstances=false --user admin_ID --password password
    nodeagentName
    ```

    ```
    AppServer8-base/bin/asadmin start-instance --port DASportNumber
         --user admin_ID --password password instanceName
    ```

    The default value for *DASportNumber* is `4848`.

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 283.

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 7.1 Guide,* `http://docs.sun.com/doc/819-4670`.

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 4 Access Manager Log and Debug Files" on page 272). Rollback is too difficult to be practical.

Multiple Instance Upgrades

In some deployment architectures Access Manager is deployed on multiple computer systems to provide for high availability and scalability.

It is usually desirable to upgrade the Access Manager instances sequentially without interrupting service. The procedure for performing a rolling upgrade of Release 2 Access Manager to Release 5 is based on the following constraint: Release 5 Access Manager can *not* co-exist with the Release 2 directory structure. However, if Directory Server instances are replicated, as in Figure 14-2, then you can perform a rolling upgrade as documented in "Multiple Instance Upgrades" on page 291.

Chapter 15

Portal Server

This chapter describes how to upgrade Portal Server to Java ES 5 (Release 5): Sun Java System Portal Server 7.1.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. 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 *PortalServer6-base and PortalServer6Config-base* (Portal Server 6.*x*) and *PortalServer7-base and PortalServer7Config-base* (Portal Server 7.*x*). 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 values of these directory paths are shown in the following table.

Table 15-1 Portal Server Directory Paths

Path Name Variable	Solaris OS	Linux OS
PortalServer6-base	`/opt/SUNWps`	`/opt/sun/portal`

Table 15-1 Portal Server Directory Paths *(Continued)*

Path Name Variable	Solaris OS	Linux OS
PortalServer6Config-base	`/etc/opt/SUNWps`	`/etc/opt/sun/portal`
PortalServer7-base	`/opt/SUNWportal`	`/opt/sun/portal`
PortalServer7Config-base	`/etc/opt/SUNWportal`	`/etc/opt/sun/portal`
PortalServer7Data-base	`/var/opt/SUNWportal`	`/var/opt/sun/portal`

Overview of Portal Server Upgrades

This section describes the following general aspects of Portal Server that impact upgrading to Java ES 5 (Release 5):

- About Java ES Release 5 Portal Server
- Portal Server Upgrade Roadmap
- Portal Server Data
- Portal Server Upgrade Strategy

About Java ES Release 5 Portal Server

Java ES Release 5 Portal Server represents a major release with respect to Release 4, with many new enhancements and features. Many of these changes were made in an Interim Feature Release (IFR) subsequent to Release 4. Release 5 represents only minor feature changes with respect to the IFR. For information about the IFR enhancements and new features, see the *Sun Java System Portal Server 7.1 Release Notes*, `http://docs.sun.com/doc/819-4986/6n4l3f365?a=view.` In particular, the Release 4 command line administrative interface has been replaced by the `psadmin` command.

Portal Server Upgrade Roadmap

Table 15-2 shows the supported Portal Server upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 15-2 Upgrade Paths to Java ES 5 (Release 5): Portal Server 7.1

Java ES Release	Portal Server Version	General Approach	Reconfiguration Required
Interim Feature Release (IFR)	Sun Java System Portal Server IFR 7.0 2005Q4	Direct upgrade: Performed by applying patches and then using an upgrade script.	Customizations need to be re-applied manually.
Release 4	Sun Java System Portal Server 6.3.1 2005Q4	Direct upgrade: Performed using an upgrade script.	Customizations need to be re-applied manually.

Table 15-2 Upgrade Paths to Java ES 5 (Release 5): Portal Server 7.1 *(Continued)*

Java ES Release	Portal Server Version	General Approach	Reconfiguration Required
Release 3	Sun Java System Portal Server 6.3.1 2005Q1	Direct upgrade: Performed using an upgrade script.	Customizations need to be re-applied manually.
Release 2	Sun Java System Portal Server 6.3 2004Q2	Direct upgrade: Performed using an upgrade script.	Customizations need to be re-applied manually.
Release 1	Sun ONE Portal Server 6.2 (2003Q4)	No direct upgrade: But can be performed by upgrading first to Release 3 and then upgrading from Release 3 to Release 5.	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 15-3 Portal Server Data Usage

Type of Data	Location	Usage
Configuration data	*PortalServer6Config-base*/	Configuration of Portal Server.
Web container access control and configuration files	Web Server 7.0 (Java ES Release 5) `server.policy` and `server.xml` files in *WebServer7Config-base*`/https-`*configName*`/config` Web Server 6.*x* (Java ES Release 2, 3, and 4) `server.policy` and `server.xml` files in *WebServer6-base*`/https-`*hostname*`/config` Application Server 8.*x* (Java ES Release 3, 4, and 5): `server.policy` and `domain.xml` files in *AppServer8Config-base*`/domains/`*domainName*`/config` Application Server 7.*x* (Java ES Release 2): `server.policy` and `server.xml` files in *AppServer7Config-base*`/domains/`*domainName*`/config`	Configuration of Portal Server web container instance.

Table 15-3 Portal Server Data Usage *(Continued)*

Type of Data	Location	Usage
Customization data	*PortalServer6Config-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.
Dynamic application data	None	Portal Server does not persistently store application data such as session state.

Portal Server Upgrade Strategy

Your strategy for upgrading Portal Server generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Portal Server by presenting issues that might influence your Portal Server upgrade plan.

Compatibility Issues

Release 5 Portal Server introduces public interface changes in the `psadmin` command used to administer Portal Server and Portal Server Secure Remote Access components. See the *Sun Java System Portal Server 7.1 Command-Line Reference*, `http://docs.sun.com/doc/819-5030`.

Hence, Release 5 Portal Server is not backwardly compatible with earlier versions, or with earlier versions of Portal Server Secure Remote Access components (including the SRA Gateway, the Rewriter Proxy, and the Netlet Proxy), except for a transitional period in which multi-instance deployments are undergoing a rolling upgrade. All Portal Server instances need to be synchronized, along with Portal Server Secure Remote Access component instances, at Java ES Release 5.

Also, individual Portal Server components, including the mobile access component, are not backwardly compatible with earlier versions; all need to be synchronized to Java ES Release 5.

In addition, there is an incompatibility between the Directory Server data structures used by Release 5 Portal Server and earlier Portal Server versions. This incompatibility impacts a rolling upgrade of multiple Portal Server instances using the same Directory Server data.

Portal Server Dependencies

Portal Server dependencies on other Java ES components can impact your 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 1-9 on page 47).
- **Web Container.** Portal Server has a mandatory dependency on web container services, which can be provided either by Java ES Web Server, Java ES Application Server, or by third-party web containers from Weblogic and WebSphere.

NOTE Upgrade of Portal Server to Release 5 is not supported for deployments in third-party web containers. For deployments in web containers from Weblogic and WebSphere, you must perform a fresh installation of Release 5 Portal Server.

- **Access Manager (or Access Manager SDK).** Portal Server has a mandatory dependency on 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 has a mandatory dependency on Directory Server, which stores user data accessed by way of Access Manager. As a result, Portal Server upgrades might require extensions of directory schema.
- **Portal Server Secure Remote Access.** Portal Server has an optional dependency on Portal Server Secure Remote Access, which provides secure remote access through the Gateway, Rewriter Proxy, and Netlet Proxy components.
- **Java DB.** Portal Server has an optional dependency on Java DB, which provides support for several portlet applications.

- **Service Registry.** Portal Server has a mandatory dependency on Service Registry, which provides libraries needed for compilation.
- **Communications Express.** Portal Server has an optional dependency on Communications Express, a Sun Java Communications Suite component, which is used to provide messaging and calendar channels to end users. Communications Express is no longer a Java ES product component.

Selective Upgrade Issues

While, in general, Java ES Release 5 supports selective upgrade of all components on a computer, the fact that Portal Server has dependencies on so many other Java ES components makes it very difficult to certify arbitrary combinations of components across various Java ES release versions.

For this reason, Portal Server supports a restricted set of upgrade scenarios with respect to Access Manager and web containers.

- **If you are upgrading Portal Server from Java ES Release 4.** You can either upgrade Directory Server, Access Manager, and web container (Web Server or Application Server) to Release 5 before upgrading Portal Server, or you can upgrade *only* Portal Server to Release 5 (leaving the other components at their Release 4 levels), but you cannot leave some dependencies at Release 4 and upgrade others to Release 5.
- **If you are upgrading Portal Server from Java ES Release 3.** You have to upgrade Directory Server, Access Manager, and web container (Web Server or Application Server) to Release 4 or to Release 5 before upgrading Portal Server, but you cannot leave any dependencies at Release 3, nor upgrade some dependencies to Release 4 and others to Release 5.
- **If you are upgrading Portal Server from Java ES Release 2.** You have to upgrade Directory Server, Access Manager, and web container (Web Server or Application Server) to Release 4 or to Release 5 before upgrading Portal Server. You cannot leave any dependencies at Release 2, nor upgrade some dependencies to Release 4 and others to Release 5.

Web Container Upgrade Scenarios

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

Table 15-4 Web Container Upgrade Scenarios for Portal Server Upgrade

Scenario	Web Container in which Portal Server is Originally Deployed	Web Container in which Portal Server is Deployed After Upgrade	Applicable Portal Server Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 6.*x*	Release 2 Release 3 Release 4 IFR 7.0
Scenario 2	Web Server 6.*x*	Web Server 7.0	Release 2 Release 3 Release 4
Scenario 3	Application Server 8.1	Application Server 8.1	Release 3 Release 4 IFR 7.0
Scenario 4	Application Server 8.1	Application Server 8.2	Release 3 Release 4 IFR 7.0
Scenario 5	Application Server 7*x*	Application Server 8.2	Release 2

You must be careful when upgrading Portal Server (for example when using the psupgrade script) to provide values appropriate to the upgrade scenario in Table 15-4 that applies, especially when there is a major version upgrade of the web container.

Dual Upgrade

Dual upgrades, in which both Portal Server and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed using the in-place operating system upgrade approach:

1. Back up existing Portal Server data.

   See "Portal Server Data" on page 312 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Portal Server.

   See the appropriate section of this chapter, depending on upgrade path.

Upgrading Portal Server from Java ES Release 4

This section includes information about upgrading Portal Server from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5).

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 or 4 prior to being upgraded to Release 5. The aforementioned upgrade path is not currently supported.

The section covers the following topics:

- Introduction
- Release 4 Portal Server Upgrade
- Multiple Instance Upgrades

Introduction

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

- **General Upgrade Approach.** The upgrade is performed using an upgrade script, `psupgrade`. The script installs new packages, migrates configuration data when necessary, updates localization files, and re-deploys Portal Server web applications to the web container.
- **Upgrade Dependencies.** Portal Server has dependencies on a number of Java ES shared components (see Table 1-9 on page 47). While Release 5 Portal Server is compatible with the Release 4 version of these shared components, upgrade of shared components is nevertheless necessary because the `psupgrade` script used to upgrade Portal Server requires the Release 5 version of the ANT shared component.

  Release 5 Portal Server also has dependencies upon a web container, Access Manager, and Directory Server, as described in "Portal Server Dependencies" on page 314. Two approaches to upgrading these dependencies are supported (see "Selective Upgrade Issues" on page 315):

  - All dependencies satisfied by Release 4 components (*none* are upgraded to Release 5)
  - All dependencies satisfied by Release 5 components (*all* are upgraded to Release 5).

- **Backward Compatibility.** Release 5 Portal Server is not backwardly compatible with the Release 4 version.
- **Upgrade Rollback.** Rollback of the Release 5 upgrade of Portal Server to Release 4 consists of restoring Release 4 packages, restoring Release 4 Directory data, and redeploying Portal Server web applications to the web container.
- **Platform Issues.** The general approach for upgrading Portal Server is the same on both Solaris and Linux operating systems, however release 5 Portal Server is installed in a new path on Solaris OS, but in the same Release 4 path on Linux OS.

Release 4 Portal Server Upgrade

This section describes how to perform an upgrade of Portal Server from Java ES Release 4 to Java ES Release 5 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:

- Release 4 Pre-Upgrade Tasks
- Upgrading Release 4 Portal Server (Solaris)
- Upgrading Release 4 Portal Server (Linux)
- Verifying the Upgrade
- Release 4 Post-Upgrade Tasks
- Rolling Back the Upgrade (Solaris)
- Rolling Back the Upgrade (Linux)

Release 4 Pre-Upgrade Tasks

Before you upgrade Portal Server, you should perform the following tasks:

- Verify Current Version Information
- Upgrade Portal Server Dependencies
- Obtain Required Configuration Information and Passwords
- Back Up Release 4 Portal Server Configuration Information
- Record Java Virtual Machine (JVM) Settings
- Remove Configuration for Load Balancer
- Remove Configuration for Directory Proxy Server

Verify Current Version Information

You can verify the current version of Portal Server using the following command:

```
PortalServer6-base/bin/version
```

Table 15-5 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]
IFR Release	7.0
Release 5	7.1

[1] The only difference between Release 3 and Release 4 is a patch. You can check for the Release 4 patches using the Solaris `showrev -p | grep` *patch_ID* command and the Linux `rpm -qa sun-portal-core` command and comparing the versions to those listed in the Java ES Release 4 *Upgrade Guide*.

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 5.

While Release 5 Portal Server is compatible with the Release 4 version of Java ES shared components, upgrade of shared components is nevertheless necessary because the `psupgrade` script used to upgrade Portal Server requires the Release 5 version of the ANT shared component.

If you choose to upgrade any of the Portal Server product component dependencies to Release 5, they *all* need to be upgraded (see “Selective Upgrade Issues” on page 315). The dependencies should be upgraded in the order below (skipping any that might already have been upgraded), before you upgrade Portal Server.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in “Upgrading Java ES Shared Components” on page 63.

2. **Directory Server.** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, “Directory Server” on page 99.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, “Web Server” on page 133 and Chapter 11, “Application Server” on page 205, 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 5 are provided in Chapter 14, “Access Manager” on page 261.

5. **Portal Server Secure Remote Access.** Instructions for upgrading Portal Server Secure Remote Access to Release 5 are provided in Chapter 16, “Portal Server Secure Remote Access” on page 379.

6. **Java DB.** Instructions for upgrading Java DB to Release 5 are provided in Chapter 8, “Java DB” on page 159.

7. **Service Registry.** Instructions for upgrading Service Registry to Release 5 are provided in Chapter 12, “Service Registry” on page 233.

8. **Communications Express.** Instructions for upgrading Communications Express to Release 5 are provided in the *Sun Java Communications Suite Upgrade Guide*, `http://docs.sun.com/doc/819-7561`.

Obtain Required Configuration Information and Passwords

Depending on the web container upgrade scenario (see Table 15-4 on page 316), the `psupgrade` script requires you to input information about passwords and other web container configuration data. The information required for different web container upgrade scenarios is shown in Table 15-6. Be sure to assemble the relevant information before beginning the Portal Server upgrade.

Table 15-6 Information Required by `psupgrade` Script per Web Container Upgrade Scenario

Information	Upgrade Scenario[3]	Web Server 7.*x* Example Values: Scenario 2	Application Server 8.*x* Example Values: Scenario 5[2]
Upgrade Portal Server on Web Server 7.0 (yes/no)	2	Yes	N/A
Web Container Install Directory	2 and 5	*WebServer7-base*	*AppServer8Install-base*
Web Container Virtual Server Instance Name	2	`https-`*configName*[4]	N/A
Web Container Instance Name	5	N/A	`server1`
Web Container Instance Directory	2	*WebServer7Config-base*/ `https-`*configName*[4]/	N/A
Portal Instance Deploy Directory	5	N/A	*AppServer8Config-base*/ `domains`/*domainName*
Web Container Instance Port	2 and 5	`80`	`80`
Web Container Instance Protocol	2 and 5	`http`	`http`
Web Container Config Name	2	*configName*[4]	N/A
Web Container Domain Name	5	N/A	`domain1`
Web Container Docs Root Directory	2 and 5	*WebServer7Config-base*/ `https-`*configName*[4]`/docs/`	*AppServer8Config-base*/ `domains`/*domainName*/ `docroot`
Web Container Admin Hostname	2 and 5	localhost	localhost
Web Container Admin Port	2 and 5	`8989`	`4848`
Web Container Admin Protocol	2 and 5	`https`	`https`
Web Container Admin User ID	2 and 5	`admin`	`admin`
Web Container Admin Password	2 through 5		
Web Container Master Password	3 through 5	N/A	
Directory Manager (cn=Directory manager) Password	1 through 5		
SRA Log User Password[1]	1 through 5		

Table 15-6 Information Required by `psupgrade` Script per Web Container Upgrade Scenario *(Continued)*

Information	Upgrade Scenario[3]	Web Server 7.*x* Example Values: Scenario 2	Application Server 8.*x* Example Values: Scenario 5[2]
Access Manager Admin Password	1 through 5		
Directory Server ldapuser Password	1 through 5		
Portal Instance ID[2]	1 through 5		

[1] This information is needed to configure Portal Server Secure Remote Access components when installed with Portal Server.

[2] A unique, non-null, value must be provided for this parameter. Values must be alpha numeric, and can include a hyphen (-).

[3] Web Container Upgrade Scenario #5 applies to upgrading Portal Server from Release 2.

[4] The default value of *configName* is *hostName.domainName.*

Back Up Release 4 Portal Server Configuration Information

Upgrade of Portal Server to Release 5 does not require the reconfiguration of Portal Server software. However, as a safety measure the `psupgrade` script will back up the following directories where configuration information is stored:

PortalServer6Config-base/

NOTE It is advisable to back up the Directory Server instance in which Portal Server stores user profiles and other data. Without such data it is not possible to roll back the upgrade to Release 5 Portal Server.

Record Java Virtual Machine (JVM) Settings

Please record the following web container JVM settings, if different from the default values, before upgrading Portal Server:

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

The location of the JVM settings depends on web container, as indicated in the following table.

Table 15-7 Location of JVM Settings

Web Container	Configuration File
Release 2, 3, 4 Web Server (6.x)	*WebServer6-base*`/https-`*instanceName*`/config/server.xml`
Release 5 Web Server (7.0)	*WebServer7Config-base*`/https-`*configName*`/config/server.xml`
Release 3, 4, 5 Application Server (8.x)	*AppServer8Config-base*`/domains/`*domainName*`/config/domain.xml`
Application Server Instance managed by Node Agent	*AppServer8Config-base*`/nodeagents/`*nodeagentName*`/config/domain.xml`

You will need to check later that these JVM settings have not been changed as a result of the Portal Server upgrade procedure.

Remove Configuration for Load Balancer

In cases in which Portal Server instances are accessed through a load balancer, the value of the `LOAD_BALANCER_URL` property used to configure such access can interfere with Portal Server upgrade. This setting must therefore be modified before performing the upgrade. To modify the `LOAD_BALANCER_URL` property setting:

1. Note which of the following configuration files are locally resident (some of which support Portal Server Secure Remote Access components that might be locally installed):

   PortalServer6Config-base`/PSConfig.properties`
   PortalServer6Config-base`/GWConfig.properties` (if Gateway is local)
   PortalServer6Config-base`/RWPConfig.properties` (if Rewriter Proxy is local)
   PortalServer6Config-base`/NLPConfig.properties` (if Netlet Proxy is local)

2. Record the current value of the `LOAD_BALANCER_URL` property in these configuration files.

3. Modify the value of the `LOAD_BALANCER_URL` property to point to the relevant Portal Server instance:

   `LOAD_BALANCER_URL=`*portalHostName*`:`*port*`/portal`

Remove Configuration for Directory Proxy Server

In cases in which Portal Server instances access Directory Server through a Directory Proxy Server instance, the Directory Proxy Server host and port number settings must be modified before performing the upgrade and then restored to their original values after upgrade is complete.

To modify the appropriate settings:

1. Record the current value of the `DS_HOST` and `DS_PORT` properties in the following Access Manager configuration file:

   AccessManagerConfig-base`/config/AMConfig.properties`

2. Modify the value of the `DS_HOST` and `DS_PORT` properties to point directly to the relevant Directory Server instance.

Upgrading Release 4 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 Release 5 takes into account the following considerations:

- All Portal Server instances corresponding to the same installed Portal Server image are upgraded at the same time.

- Portal Server software consists of subcomponents that perform a number of different roles, but are all upgraded together:

  - **Portal-base.** Includes administrative Mbeans and accompanying administrative software, Logging Framework, and monitoring-related software, all of which are packaged together.

  - **Portal Server web applications.** Consists of a number of web applications that are deployed in a web container. At least some of these web applications require support from Access Manager and, in turn, Directory Server.

  - **Secure Remote Access core.** Software that supports Portal Server Secure Remote Access: some servlets and applets embedded in jar files and some supporting files that cannot be deployed in a web container.

- The `psupgrade` script automatically detects which Portal Server subcomponents and which web container dependencies are installed on the host computer. For example, the script queries the system to detect the version of Application Server or Web Server to which you are deploying Portal Server web applications, and tailors the information it requests depending on the information it can detect.

Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server 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, synchronize all shared components to Release 5.

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

   This step is a necessary prerequisite to running the `psupgrade` script in Step 8 on page 326.

3. Stop any instances of the Portal Server Secure Remote Access Gateway, Rewriter Proxy, or Netlet Proxy that might be running locally.

   PortalServer6-base`/bin gateway stop`
   PortalServer6-base`/bin netletd stop`
   PortalServer6-base`/bin rwproxyd stop`

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

4. Make sure Access Manager is running if it is deployed to a web container different from the one to which Portal Server is deployed.

5. If not already running, start Portal Server by starting the web container to which it is deployed.

   Web Server 6.*x:*
   WebServer-base`/https-`*instanceName*`/start`

Web Server 7.0*:*
Admin Server--
WebServer7Config-base`/admin-server/bin/startserv`
Instance Server--
WebServer7Config-base`/https-`*configName*`/bin/startserv`

Application Server *8.x:*

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
      --password password domainName
```

6. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script. For example,

```
export ANT_HOME=/usr/sfw
export JAVA_HOME=/usr/jdk/entsys-j2se
```

7. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

8. Run the `psupgrade` script from the Java ES Release 5 distribution.

```
cd os_arch/Products/portal_svr/Tools/upgrade/bin
./psupgrade
```

where *os_arch* matches your platform, such as `Solaris_sparc`.

NOTE If you inadvertently run `psupgrade` from the wrong *os_arch* directory, you need to back out the procedure as follows:

a. Change to the correct *os_arch* directory.

b. Reverse changes to Portal Server data.

```
./psupgrade rollback
```

Provide requested parameters and passwords.

c. Run `psupgrade` once again.

The `psupgrade` script detects installed Portal Server components and localization packages, invokes the Java ES installer to install new packages, and queries the system to detect the location and port number and other information regarding the web container to which you are deploying Portal

Server web applications. Depending on web container upgrade scenario (see Table 15-4 on page 316), the script requests you to input additional information required to deploy Portal Server to the appropriate web container.

Table 15-6 on page 321 shows the information requested for the different web container upgrade scenarios in Table 15-4.

NOTE Be sure you enter correct values for psupgrade parameters, as you can't go back and change them, and it is also very difficult to roll back changes made by the psupgrade script. To reverse changes to Portal Server data, you have to run

```
./psupgrade rollback
```

before trying to run psupgrade again.

9. If necessary, restore web container JVM settings.

   To make sure that JVM settings support Release 5 Portal Server, perform the following steps:

   a. Check that the web container JVM settings for Portal Server that you recorded before upgrade have not changed as a result of the upgrade procedure.

      See "Record Java Virtual Machine (JVM) Settings" on page 322.

   b. If the settings have changed, restore them to the values you recorded before upgrade.

      Make sure the following JVM settings are included even if they were not previously set:

      ```
      <jvm-options>-XX:MaxPermSize=256m</jvm-options>
      <jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
      <jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
      ```

10. 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.

a. Stop the web container as follows:

Web Server 6.*x:*
WebServer-base/https-*instanceName*/stop

Web Server 7.0*:*
Admin Server--
WebServer7Config-base/admin-server/bin/stopserv
Instance Server--
WebServer7Config-base/https-*configName*/bin/stopserv

Application Server *8.x:*
AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
    --password *password domainName*

b. Restart the web container using the commands in Step 5 on page 325.

Upgrading Release 4 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 Release 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 324), except that Release 5 Portal Server is installed in the same path as Release 4 on Linux OS. As a result, the `psupgrade` script removes the previous RPMs when installing the Release 5 RPMs.

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 4 to Release 5 on Linux cannot be rolled back. Make sure you back up your system *before* performing the following procedure.

1. Log in as root or become superuser.

```
su -
```

2. **If you have not already done so, synchronize all shared components to Release 5.**

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

   This step is a necessary prerequisite to running the `psupgrade` script in Step 8 on page 330.

3. **Stop any instances of the Portal Server Secure Remote Access Gateway, Rewriter Proxy, or Netlet Proxy that might be running locally.**

   PortalServer6-base`/bin gateway stop`
   PortalServer6-base`/bin netletd stop`
   PortalServer6-base`/bin rwproxyd stop`

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

4. **Make sure Access Manager is running if it is deployed to a web container different from the one to which Portal Server is deployed.**

5. **If not already running, start Portal Server by starting the web container to which it is deployed.**

   Web Server 6.*x:*
   WebServer-base`/https-`*instanceName*`/start`

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base`/admin-server/bin/startserv`
   Instance Server--
   WebServer7Config-base`/https-`*configName*`/bin/startserv`

   Application Server *8.x:*
   AppServer8-base`/bin/asadmin start-domain --user` *admin_ID*
   `--password` *password domainName*

6. **Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script. For example,**

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

7. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

8. Run the psupgrade script from the Java ES Release 5 distribution.

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   where *os_arch* matches your platform, such as Linux_x86.

   The psupgrade script detects installed Portal Server components and localization packages, invokes the Java ES installer to install new packages, and queries the system to detect the location and port number and other information regarding the web container to which you are deploying Portal Server web applications. Depending on web container upgrade scenario (see Table 15-4 on page 316), the script requests you to input additional information required to deploy Portal Server to the appropriate web container.

   Table 15-6 on page 321 shows the information requested for the different web container upgrade scenarios in Table 15-4.

   NOTE Be sure you enter correct values for psupgrade parameters, as you can't go back and change them, and it is also very difficult to roll back changes made by the psupgrade script. Reminder: back up your system *before* running the psupgrade script.

9. Modify the *PortalServer7Config-base*/platform.conf.default configuration file.

   Copy the line with gateway.logging.password from the following file, which was backed up by psupgrade:

   PortalServer6Config-base.bak/platform.conf.default

   and place the line in *PortalServer7Config-base*/platform.conf.default.

10. If necessary, restore web container JVM settings.

    To make sure that JVM settings support Release 5 Portal Server, perform the following steps:

    a. Check that the web container JVM settings for Portal Server that you recorded before upgrade have not changed as a result of the upgrade procedure.

       See "Record Java Virtual Machine (JVM) Settings" on page 322.

b. If the settings have changed, restore them to the values you recorded before upgrade.

Make sure the following JVM settings are included even if they were not previously set:

```
<jvm-options>-XX:MaxPermSize=256m</jvm-options>
<jvm-options>-XX:+CMSPermGenSweepingEnabled</jvm-options>
<jvm-options>-XX:+CMSClassUnloadingEnabled</jvm-options>
```

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.

a. Stop the web container as follows:

Web Server 6.*x:*
WebServer-base`/https-`*instanceName*`/stop`

Web Server 7.0*:*
Admin Server--
WebServer7Config-base`/admin-server/bin/stopserv`
Instance Server--
WebServer7Config-base`/https-`*configName*`/bin/stopserv`

Application Server *8.x:*
AppServer8-base`/bin/asadmin stop-domain --user` *admin_ID*
`--password` *password domainName*

b. Restart the web container using the commands in Step 5 on page 329.

Verifying the Upgrade

You can verify the installation of Release 5 packages using the following command:

PortalServer7-base`/bin/psadmin --version --adminuser` *admin_ID*
`-f` *adminpasswordfile*`.`

See Table 15-5 on page 319 for output values.

To verify the full upgrade, confirm that the Portal Desktop comes up and the `psadmin` administration utility functions as documented.

You can also check the following upgrade log files:

`/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrade.log*`
PortalServer7Data-base`/logs/admin/`
PortalServer7Data-base`/logs/config/`

Release 4 Post-Upgrade Tasks

Please note the post-upgrade procedures required to address the following situations:

- Migrate Custom web-src Data
- Redeploy Custom Portlet Applications
- Migrate Customized Portlet Applications
- Correct Links in Bookmark and Application Channels
- Correct Access to Search Server
- Restore Configuration for Directory Proxy Server
- Migrate Custom web-src Data
- Manually Register Portal Server Secure Remote Access Components
- Enable the URLScrapper Channel
- Change in Logout Page

Migrate Custom web-src Data

If you have added custom data, such as images, javascript files or any other files for constructing `portal.war` to the following directory:

PortalServer6-base`/web-src`

you have to copy these additional files to the corresponding directory in Release 5 Portal Server:

PortalServer7-base`/web-src`

Redeploy Custom Portlet Applications

If you have created and deployed custom portlet applications, then these portlets must be manually redeployed after upgrade to Release 5 Portal Server. Even though display profile entries will exist and the channel name will be displayed, content will not be seen until you redeploy your custom portlets.

Redeploy portlets using the following command:

PortalServer7-base`/bin/psadmin deploy-portlet`

You can confirm redeployment by looking for the corresponding `.war` and XML files in the following location:

PortalServer7Data-base`/portals/Upgraded/war`

Migrate Customized Portlet Applications

Portlet applications based on the user interface framework provided by Sun Java Web Console (SJWC) need to be manually migrated to Release 5 and redeployed.

In particular, this requirement applies to four web applications distributed with Portal Server as sample portlet applications meant to be customized and installed into a portal: `filesharing`, `surveys`, `wiki`, `rssportlet`. During upgrade, bug-fixed versions of these sample portlet applications are placed on disk in the portlet applications area. If you have customized these portlet applications for your own use, you need to manually migrate them to Release 5 and redeploy them; they are not handled by the Portal Server upgrade process.

By default, some of these portlet applications (`filesharing` and `surveys`) are deployed and available for users when a community is created.

You can upgrade, customize, and redeploy SJWC-based portlet applications using the following procedure. The `filesharing` portlet application is used as an example:

1. Extract the Release 5 SJWC jar files.

   a. `mkdir /tmp/lh`

   b. `cd /tmp/lh`

   c.
   ```
   /usr/jdk/entsys-j2se/bin/jar xvf
      PortalServer7-base/portlet/communityportlets.war
      WEB-INF/lib/commons-beanutils.jar
      WEB-INF/lib/commons-collections-3.1.jar
      WEB-INF/lib/commons-digester.jar
      WEB-INF/lib/commons-logging.jar
      WEB-INF/lib/dataprovider.jar WEB-INF/lib/jsf-api.jar
      WEB-INF/lib/jsf-impl.jar WEB-INF/lib/webui.jar
   ```

   d. Rename one of the files.

   ```
   mv WEB-INF/lib/commons-collections-3.1.jar
   WEB-INF/lib/commons-collections.jar
   ```

2. Locate the `filesharing` portlet application.

   ```
   cd PortalServer7Config-base/portals/portal1/portletapps/filesharing
   ```

3. Inject the updated SJWC libraries.

   ```
   jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF
   ```

4. Customize the `filesharing` portlet application.

   ```
   ant customize
   ```

5. Redeploy the `filesharing` portlet application.

   a. *PortalServer7-base*`/bin/psadmin undeploy-portlet -u` *amadmin* `-f passwordfile -p` *portal_id* `-i` *instance_id* `-g filesharing`

   b. `ant deploy`

   c. Go to the following directory (depending on web container):

      Application Server 8.x:

      AppServer8Config-base`/domains/domain1/applications/j2ee-modules/communityportlets/WEB-INF`

      Web Server 6.x:

      WebServer6-base`/https-`*instanceName*`/webapps/https-`*instanceName*`/communityportlets/WEB-INF`

      Web Server 7.x:

      WebServer7Config-base`/https-`*configName*`/web-app/https-`*configName*`/communityportlets/WEB-INF`

   d. Open the `sun-web.xml` file and add the following line just before the last line (that is, before the `sun-web-app` end tag):

      ```
      <class-loader delegate="false"/>
      ```

   e. Repeat Step c and Step d for the `sun-web.xml` file under `filesharing/WEB-INF`.

6. Repeat Step 2 through Step 5 for `surveys` and any other custom portlet application based on the SJWC framework.

7. Restart the web container.

Correct Links in Bookmark and Application Channels

When upgrading Release 4 Portal Server to Release 5, the bookmark and application channels have duplicate and spurious links. To fix these links, perform the following procedure.

1. Log in to PSConsole.

2. From the Common Tasks tab, click on Manage Channels & Containers.

3. Choose DeveloperSample [Org] as DN and click OK.

4. Select JSPTabContainer [default] as the View Type.

5. Under MyFrontPageTabPanelContainer, click on App Channel.

   App channel properties will be displayed on the right-hand side frame.

   To view properties for a specific locale, Click on Table Preferences and provide the Locale value: `de`, `fr`, `es`, `ja`, `ko`, `zh`, `zh_CN`, `zh_TW`.

6. Edit the userApps property.

   a. In userApps property click on [Edit Values...] link

      A pop-up window with existing applications will appear.

   b. Remove the following applications from the list:

      NetMail Lite
      NetMail

   c. Add following application to the list:

      NetFile

   d. Click on Save and then Close.

7. Edit the target property.

   a. In target property click on [Edit Values...] link

      A pop-up window with existing targets will appear.

   b. Remove the following targets from the list:

      ```
      NetMailLite|
      NetMailServlet?nsid=newHTMLSessionNetMailLite|
      NetMailServlet?nsid=newHTMLSession
      ```

      ```
      NetMail|NetMailServlet?nsid=newAppletSession
      ```

   c. Remove the duplicate occurrence of the following targets:

      ```
      Instant Messenger (Java WebStart)|
      IMLaunch?provider=IMChannel&launch=jnlp&last=false
      ```

      ```
      Instant Messenger (Browser)|
      IMLaunch?provider=IMChannel&launch=plugin&last=false
      ```

   d. Add the following target to the list:

      ```
      NetFile|/portal/NetFileApplet?Refer=java2
      ```

   e. Click on Save and then Close.

Correct Access to Search Server

When upgrading Release 4 Portal Server to Release 5, the search server is separated from Portal Server and URL access to the search server is therefore changed.

- In Release 4 the URL is `http://`*hostName*`:`*port*`/portal/search`
- In Release 5 the URL is `http://`*hostName*`:`*port*`/UpgradedSearch/search`

As a result, you have to manually modify the Display Profiles for all portal channels that implement the SearchProvider or DiscussionProvider interfaces, such as Search, DiscussionLite, Discussions, and Instant Messaging Channel. In particular, you have to modify the `searchServer` property of these channels, at whatever organizational or role level they might occur within the Display Profile, to correctly reference the search server. Modify the `searchServer` value as follows:

```
value="http://hostName:port/UpgradedSearch/search"
```

Also, for the Instant Messaging Channel, the Instant Messaging Server configuration property, `iim_arch.portal.search`, needs to be updated with the new search server URL.

Restore Configuration for Directory Proxy Server

If Portal Server instances have accessed Directory Server through a Directory Proxy Server instance, the Directory Proxy Server host and port number settings must be restored to their original values before upgrade. See "Remove Configuration for Directory Proxy Server" on page 324, in which the values of these properties was modified in preparation for upgrade.

Manually Register Portal Server Secure Remote Access Components

If you encounter a Null Pointer exception issue reported to the standard output at the end of the upgrade procedure, it means registration of Portal Server Secure Remote Access components, if any, has failed.

In this situation, you can manually register (enable) the Portal Server Secure Remote Access components by executing the following command:

```
PortalServer7-base/bin/psadmin provision-sra -u amadminUser -f passwordFile
    -p Portal_ID --gateway-profile profileName --enable
```

Enable the URLScrapper Channel

When upgrading from Release 4 to Release 5, you have to enable the URLScrapper channel. Use the following procedure:

1. Log in to Portal Server Console

   Click on Portal tab and then click on upgraded portal.

2. From the Select DN drop down menu, select TopLevel (Global) and click on Download Display Profile link.

   Store the downloaded file in some temporary location

3. Locate `com.sun.portal.providers.urlscraper.URLScraperProvider`.

4. Locate the XML portion starting with:

   ```
   <Provider advanced="false"
   class="com.sun.portal.providers.urlscraper.URLScraperProvider"
   ```

   and ending with:

   ```
   </Provider>
   ```

5. Replace the XML portion in Step 4 with the following:

   ```
   <Provider advanced="false"
   class="com.sun.portal.providers.urlscraper.URLScraperProvider"
   container="false" lock="false" merge="fuse"
   name="URLScraperProvider" version="2">

   <Properties advanced="false" lock="false" merge="fuse"
   name="_properties" propagate="true">

   <String advanced="false" lock="false" merge="replace" name="title"
   propagate="true" value="UrlScraper Channel"/>

   <String advanced="false" lock="false" merge="replace"
   name="description" propagate="true" value="This is a test for
   urlscraper"/>

   <Boolean advanced="true" lock="false" merge="replace"
   name="isEditable" propagate="true" value="false"/>

   <Boolean advanced="true" lock="false" merge="replace"
   name="isTopLevel" propagate="true" value="false"/>

   <String advanced="true" lock="false" merge="replace" name="editType"
   propagate="true" value="edit_subset"/>

   <Boolean advanced="true" lock="false" merge="replace"
   name="enableUBT" propagate="true" value="false"/>

   <String advanced="false" lock="false" merge="replace"
   name="urlScraperRulesetID" propagate="true"
   value="default_ruleset"/>
   ```

```
<String advanced="false" lock="false" merge="replace" name="width"
propagate="true" value="thick"/>

<String advanced="true" lock="false" merge="replace"
name="refreshTime" propagate="true" value="0"/>

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

<String advanced="false" lock="false" merge="replace"
name="fontFace1" propagate="true" value="Sans-serif"/>

<String advanced="false" lock="false" merge="replace"
name="productName" propagate="true" value="Sun JavaTM System Portal
Server 7"/>

<Boolean advanced="false" lock="false" merge="replace"
name="cookiesToForwardAll" propagate="true" value="true"/>

<String advanced="false" lock="false" merge="replace"
name="inputEncoding" propagate="true" value="UTF-8"/>

<Collection advanced="false" lock="false" merge="fuse"
name="cookiesToForwardList" propagate="true"/>

<Integer advanced="false" lock="false" merge="replace"
name="timeout" propagate="true" value="100"/>

<String advanced="true" lock="false" merge="replace" name="formData"
propagate="true" value=""/>

<Boolean advanced="true" lock="false" merge="replace"
name="isHttpAuth" propagate="true" value="false"/>

<String advanced="true" lock="false" merge="replace" name="loginUrl"
propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace"
name="loginFormData" propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace" name="uid"
propagate="true" value=""/>

<String advanced="true" lock="false" merge="replace" name="password"
propagate="true" value=""/>
```

```
<ConditionalProperties advanced="false" condition="client"
lock="false" merge="fuse" propagate="true" value="HTML">

<ConditionalProperties advanced="false" condition="locale"
lock="false" merge="fuse" propagate="true" value="en">

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

</ConditionalProperties>

<String advanced="true" lock="false" merge="replace" name="helpURL"
propagate="true" value="en/desktop/urlscrpr.htm"/>

<String advanced="false" lock="false" merge="replace" name="url"
propagate="true" value=""/>

</ConditionalProperties>

<ConditionalProperties advanced="false" condition="locale"
lock="false" merge="fuse" propagate="true" value="en">

<String advanced="false" lock="false" merge="replace" name="title"
propagate="true" value="UrlScraper Channel"/>

<String advanced="false" lock="false" merge="replace"
name="description" propagate="true" value="This is a test for
urlscraper"/>

</ConditionalProperties>

</Properties>

</Provider>
```

6. Save and upload the modified file.

Change in Logout Page

The Release 5 Portal Server logout page has been changed from the previous Access Manager logout page. Please note that this change does not represent a defect in the software.

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 5 consists of reverting back to the Release 4 installation at *PortalServer6-base* and redeploying the Release 4 web applications.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Restore Directory Server to the state it was in before upgrade.

   Use the Directory Server backup/restore command line and GUI utilities. See the Directory Server Backup and Restore chapter of the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide,* http://docs.sun.com/doc/819-0995.

3. Stop Portal Server by stopping its web container.

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/stop

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/stopserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/stopserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
   --password *password domainName*

4. Remove the Release 5 Portal Server packages.

   a. Launch the Java ES uninstaller.

      ```
      /var/sadm/prod/SUNWentsys5/uninstall
      ```

   b. Select all installed Portal Server components.

   c. Confirm your uninstall choice.

   d. Exit the Java ES uninstaller.

5. Restart Portal Server by restarting its web container.

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/start

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/startserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/startserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

6. Re-deploy the Release 4 Portal Server web applications using the following command from the Java ES Release 5 distribution:

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade rollback
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`.

   The `psupgrade rollback` command un-deploys Release 5 Portal Server web applications and re-deploys Release 4 Portal Server web applications.

   The command redeploys content from *PortalServer6-base*/web-src to /var/*PortalServe6-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 `psupgrade rollback` command, or such changes will be overwritten.

7. 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.

Rolling Back the Upgrade (Linux)

Because the upgrade to Release 5 requires the removal of the Release 4 binaries, it is very difficult to roll back the upgrade on Linux.

One approach to rollback would be to create a parallel system *before* upgrading and testing that system before attempting an upgrade. If you need to roll back the upgrade, you can revert back to that parallel system.

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 instances 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, as described below. The procedure takes into account the following constraint: Release 4 Portal Server does not work with Release 5 Portal Server directory data.

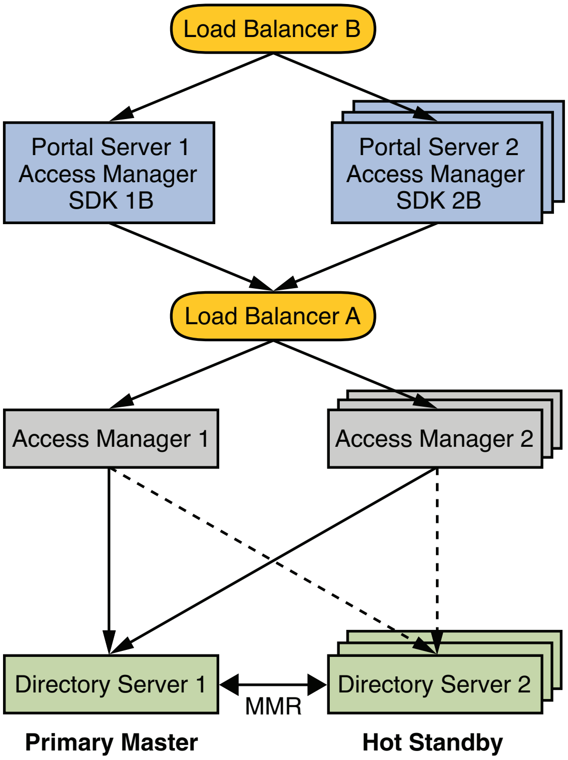
The deployment architecture shown in Figure 15-1 on page 343 will be used to illustrate the procedure for a rolling upgrade of Release 4 Portal Server instances to Release 5.

NOTE For architectures that include Portal Server Secure Remote Access components, see "Multiple Instance Upgrades" on page 399.

In the architecture of Figure 15-1, multiple Portal Server instances are accessed by way of a load balancer to provide for availability and scalability. The Portal Server instances, in turn, access Access Manager instances through a load balancer. The The Access Manager and Access Manager SDK instances access a directory that is set up for multi-master replication (MMR). While other Directory Server replication schemes are possible, MMR is representative of highly available and scalable directory services.

In Figure 15-1, the multiple instances of Portal Server, Access Manager, and Directory Server are grouped to facilitate explanation of the upgrade procedure. `Portal Server 2`, for example, is representative of the second through nth instances of Portal Server.

Figure 15-1 Example Deployment Architecture for Multiple Portal Server Instances



Rolling upgrade of Release 4 Portal Server to Release 5 is performed as follows:

1. If you are upgrading Release 4 Access Manager to Release 5, perform a rolling upgrade as documented in "Multiple Instance Upgrades" on page 285. Note that in upgrading Release 4 Portal Server to Release 5, you are not required to upgrade Release 4 Access Manager to Release 5.

2. Configure `Portal Server 2` to point to `Directory Server 2` rather than `Directory Server 1`.

   For brevity, in this and succeeding steps, "`Portal Server 2`" will mean `Portal Server 2` through `Portal Server n`.

3. Upgrade `Portal Server 1`.

   a. Disable `Portal Server 1` in `Load Balancer B`.

      Requests will no longer be routed to `Portal Server 1`.

b. Disable Directory Server MMR.

   Directory Server 2 will no longer by synchronized with Directory Server 1.

c. Upgrade Access Manager SDK 1B to Release 5.

   Use the procedure in "Release 4 Access Manager SDK-only Upgrades" on page 287.

d. Upgrade Portal Server 1 to Release 5.

   Perform the upgrade of the Portal Server instance as described in "Release 4 Portal Server Upgrade" on page 318, noting the following:

   - Make special note of the following pre-upgrade task: "Remove Configuration for Load Balancer" on page 323.
   - Confirm, before performing the upgrade, that the value of am.encryption.pwd in the *AccessManagerConfig-base*/config/AMConfig.properties file is the same for the local Access Manager SDK as for its associated remote Access Manager instance.
   - Make sure that you provide a non-null, unique value for the Portal Instance ID parameter requested by psupgrade for each Portal Server instance that you are upgrading.

   Portal Server data for Directory Server 1 is updated to Release 5.

e. Enable Portal Server 1 in Load Balancer B.

   Requests will be once again routed to Portal Server 1.

4. Upgrade Portal Server 2.

   a. Disable Portal Server 2 in Load Balancer B.

      Requests will no longer be routed to Portal Server 2.

   b. Restore the configuration of Portal Server 2 to point to Directory Server 1.

   c. Upgrade Access Manager SDK 2B to Release 5.

      Use the same procedure as in Step c on page 344.

d. Upgrade `Portal Server 2` to Release 5.

   Use the same procedure as in Step d on page 344.

e. Enable `Portal Server 2` in `Load Balancer B`.

   Requests will be once again routed to `Portal Server 2`.

5. Enable Directory Server MMR.

   The Portal Server data for `Directory Server 2`, is now synchronized with `Directory Server 1`.

Upgrading Portal Server from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Portal Server to Release 5 is the same as that for upgrading Release 4 Portal Server to Release 5, with the following exceptions:

- Release 3 Pre-Upgrade Task: Upgrading Portal Server Dependencies
- Upgrading Release 3 Portal Server
- Multiple Instance Upgrades

Release 3 Pre-Upgrade Task: Upgrading Portal Server Dependencies

However, when upgrading Portal Server from Release 3, you have to upgrade *both* Access Manager and web container (Web Server or Application Server) to Release 4 or to Release 5 before upgrading Portal Server, but you cannot leave any dependencies at Release 3, nor upgrade some dependencies to Release 4 and others to Release 5. For more information, see "Selective Upgrade Issues" on page 315.

The following dependencies need to be upgraded in the order shown below.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 5 are provided in Chapter 5, "Directory Server" on page 99.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, "Web Server" on page 133 and Chapter 11, "Application Server" on page 205, 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 5 are provided in Chapter 14, "Access Manager" on page 261.

Upgrading Release 3 Portal Server

To upgrade Release 3 Portal Server to Release 5, use the instructions in "Upgrading Portal Server from Java ES Release 4" on page 317, except substitute Release 3 wherever Release 4 is referenced.

Release 3 Post-Upgrade Tasks

When upgrading Portal Server from Release 3 to Release 5, you must perform, in addition to the post-upgrade procedures documented in "Release 4 Post-Upgrade Tasks" on page 332, the post-upgrade procedures required to address the following situations:

- Subscribing a Discussion

Subscribing a Discussion

Subscribing a discussion in a community will not succeed unless you first edit the global display profile top level properties to add the following String property:

```
helpURL=en/desktop/usedesk.htm
```

Use the following procedure:

1. Create a display profile XML snippet file, `helpUrl.xml`:

```
<?xml version="1.0" encoding="utf-8" ?>
  <!DOCTYPE DisplayProfile SYSTEM "jar://resources/psdp.dtd">
    <Properties>
      <String name="helpURL" value="en/desktop/usedesk.htm" />
    </Properties>
```

2. Run the Global display profile properties using the following command:

```
./psadmin modify-dp -u amadminUser -f /tmp/passwordFile -p portal_ID
    -m -g helpUrl.xml
```

where the `-m` option is required to not overwrite the entire Global display profile.

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 instances 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, as described below. The procedure takes into account the following constraint: Release 3 Portal Server does not work with Release 5 Portal Server directory data.

To perform a rolling upgrade from Release 3 Portal Server to Release 5, use the same procedure documented in "Multiple Instance Upgrades" on page 342, except substitute Release 3 wherever Release 4 is referenced. In addition, you must also upgrade Access Manager, as described in Step 1 on page 343.

Upgrading Portal Server from Java ES Release 2

This section includes information about upgrading Java ES 2004Q2 (Release 2) Portal Server to Release 5. The upgrade procedure is similar to that for upgrading Release 4 Portal Server to Release 5, except for some changes as documented in the following sections:

- Release 2 Pre-Upgrade Tasks
- Upgrading Release 2 Portal Server
- Release 2 Post-Upgrade Tasks
- Multiple Instance Upgrades

NOTE If you are upgrading from Release 2 Portal Server on the Linux platform, then you will have to perform a dual upgrade, in which both Portal Server *and* the operating system are upgraded (Release 5 Portal Server is not supported on RHEL 2.1). See "Dual Upgrade" on page 316 for more information.

Release 2 Pre-Upgrade Tasks

The pre-upgrade tasks for upgrading Portal Server from Release 2 are the same as those documented in "Release 4 Pre-Upgrade Tasks" on page 319, except for upgrading Portal Server dependencies.

When upgrading Portal Server from Release 2, you have to upgrade *both* Access Manager and web container (Web Server or Application Server) to Release 4 or to Release 5 before upgrading Portal Server, but you cannot leave any dependencies at Release 2, nor upgrade some dependencies to Release 4 and others to Release 5. For more information, see "Selective Upgrade Issues" on page 315.

In particular, the web container software must be upgraded from Release 2, meaning that only Scenario 2 and Scenario 5 from Table 15-4 on page 316 are supported when running the `psupgrade` script.

The following dependencies need to be upgraded in the order shown below.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 5 are provided in "Upgrading Directory Server from Java ES Release 2" on page 115.

3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in Chapter 7, "Web Server" on page 133 and Chapter 11, "Application Server" on page 205, 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 5 are provided in Chapter 14, "Access Manager" on page 261.

Upgrading Release 2 Portal Server

The procedure for upgrading Portal Server from Release 2 to Release 5 depends on the web container in which you are deploying Portal Server software, as described in the following sections.

Upgrading Release 2 Portal Server: Web Server Web Container

To upgrade Release 2 Portal Server to Release 5, when deploying into a Web Server web container, follow the instructions in "Upgrading Portal Server from Java ES Release 4" on page 317, except substitute Release 2 wherever Release 4 is referenced.

However, if Portal Server is deployed to Release 5 Web Server (Web Server 7.0), then you must perform the following steps before upgrading Release 2 Web Server:

1. Log in to the Web Server Admin Console.
2. Click on Edit Virtual Severs > Web Applications.
3. Remove all deployed web applications that have a URI that includes either `/portal` or `/portalsamples.`
4. Click on Save.
5. Click on Deployment Pending.

Upgrading Release 2 Portal Server: Application Server Web Container

When upgrading Release 2 Portal Server to Release 5, when deploying into an Application Server web container, the Application Server has been upgraded from Release 2 to Release 5.

The Release 2 Application Server instance in which Portal Server was originally deployed (*instanceName*), when upgraded to Release 5, was migrated under a node agent created by the Application Server upgrade process. Upgrade of Portal Server in this upgraded Application Server instance requires the following steps:

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, synchronize all shared components to Release 5.

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

   This step is a necessary prerequisite to running the `psupgrade` script in Step 9 on page 352.

3. Stop any instances of the Portal Server Secure Remote Access Gateway, Rewriter Proxy, or Netlet Proxy that might be running locally.

   ```
   PortalServer6-base/bin gateway stop
   PortalServer6-base/bin netletd stop
   PortalServer6-base/bin rwproxyd stop
   ```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

4. Make sure Access Manager is running if it is deployed to a web container different from the one to which Portal Server is deployed.

5. If not already running, start Portal Server by starting the web container to which it is deployed.

   a. Start the Domain Administration Server (DAS) if it is not already started.

      ```
      AppServer8-base/bin/asadmin start-domain --user admin_ID
           --password password domainName
      ```

b. Start the upgraded Application Server instance in which Portal Server 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:

- where *nodeagentName* has the form *hostName_domainName*, but is simply *hostName* by default.
- The default *domainName* is `domain1`
- The default *instanceName* is `server1`

6. Undeploy Portal Server components.

   ```
   AppServer8-base/bin/asadmin undeploy --user admin_ID
        --password password --target instanceName portal
   ```

   ```
   AppServer8-base/bin/asadmin undeploy --user admin_ID
        --password password --target instanceName portletsamples
   ```

7. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script. For example,

   Solaris OS:

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

   Linux OS:

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

8. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

9. Run the `psupgrade` script from the Java ES Release 5 distribution.

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`.

NOTE If you inadvertently run psupgrade from the wrong *os_arch* directory, you need to back out the procedure as follows:

a. Change to the correct *os_arch* directory.

b. Reverse changes to Portal Server data.

```
./psupgrade rollback
```

Provide requested parameters and passwords.

c. Run psupgrade once again.

The psupgrade script invokes the Java ES installer to install new packages and queries the system to detect the location and port number and other information regarding the web container to which you are deploying Portal Server web applications. Depending on web container upgrade scenario (see Table 15-4 on page 316), in this case Scenario 5, the script requests you to input information required to deploy Portal Server to the appropriate web container.

Table 15-6 on page 321 shows the information requested when Release 2 Application Server has been upgraded to Release 5 (Scenario 5).

NOTE Be sure you enter correct values for psupgrade parameters, as you can't go back and change them, and it is also very difficult to roll back changes made by the psupgrade script. To reverse changes to Portal Server data, you have to run

```
./psupgrade rollback
```

before trying to run psupgrade again.

10. Stop the Domain Administration Server (DAS) and node agent that were started in Step 5 on page 351.

```
AppServer8-base/bin/asadmin stop-domain --user admin_ID
    --password password domainName
```

```
AppServer8-base/bin/asadmin stop-node-agent --user admin_ID
    --password password nodeagentName
```

11. Restart the Domain Administration Server (DAS), node agent, and server instance that were stopped in Step 10.

NOTE Be sure to separately start the node agent using the `startinstances=false` option before starting the server instance, as shown below.

```
AppServer8-base/bin/asadmin start-domain --user admin_ID
     --password password domainName
```

```
AppServer8-base/bin/asadmin start-node-agent --port DASportNumber
     --startinstances=false --user admin_ID --password password
nodeagentName
```

```
AppServer8-base/bin/asadmin start-instance --port DASportNumber
     --user admin_ID --password password instanceName
```

The default value for *DASportNumber* is `4848`.

Release 2 Post-Upgrade Tasks

When upgrading Portal Server from Release 2 to Release 5, you must perform, in addition to the post-upgrade procedures documented in "Release 4 Post-Upgrade Tasks" on page 332, the post-upgrade procedures required to address the following situations:

- Single Sign-on Configuration
- Enabling the URLScrapper Channel
- Delete Gateway Service Entry

Single Sign-on Configuration

After upgrade of Portal Server from Release 2, Portal desktop communication channels, such as Mail, Calendar, and Address Book, that use the ssoadapter meta-template to access a back-end server will fail.

For example, if you have modified the Release 2 mail ssoadapter meta-template, `SUN-UWC-MAIL`, with settings specific to your Messaging Server, then after upgrade to Release 5, two `SUN-UWC-MAIL` ssoadapter meta-templates will exist: one is your Release 2 version, which has not been changed, and the other is the new Release 5 version. You will observe duplicate ssoadapter meta-templates in the Portal Server Console and `psadmin` command line interface, both with the same name.

Channels that use the ssoadapter meta-templates will be unable to establish a connection with the back-end server and retrieve data.

To fix this problem you have to retrieve the ssoadapter meta-template data, rename the duplicate entries, and then replace the modified data. Use the following procedure:

1. Export the ssoadapter meta-template data.

   You use the `amadmin` utility to export Access Manager service data, as follows:

   a. Create an `amadmin` request file, `/tmp/ssoadapter-template-gets.xml`.

      This file will be used by the utility to retrieve the ssoadapter meta-template data:

      ```
      <?xml version="1.0" encoding="ISO-8859-1"?>
      <!DOCTYPE Requests
        PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
        "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
        >
        <Requests>
          <SchemaRequests serviceName="SunSSOAdapterService"
          SchemaType="global">
            <GetServiceDefaultValues>
              <Attribute name="sunConfigurationTemplates" />
            </GetServiceDefaultValues>
          </SchemaRequests>
        </Requests>
      ```

   b. Execute the following asadmin command:

      ```
      AccessManager-base/bin/amadmin -u amadminUser -w password
       -t ssoadapter-templates-get.xml > /tmp/ssoadapter-templates.xml
      ```

      The output of the command is saved to `/tmp/ssoadapter-templates.xml`.

      The `/tmp/ssoadapter-templates.xml` file has the following format:

      ```
      sunConfigurationTemplates=
      [<ssoadapter meta-template1>, <ssoadapter meta-template2>, ...]
      ```

      and each `<ssoadapter meta-template>` has the following syntax:

      ```
      default|imap:/?configName=SUN-UWC-MAIL
      &proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D&subType=sun-one
      &enableProxyAuth=false ...
      ```

2. Modify the `/tmp/ssoadapter-templates.xml` file to rename the duplicate ssoadapter meta-templates.

   a. Find each template in the `/tmp/ssoadapter-templates.xml` file.

      Look for the `default|imap:/?configName=` string.

   b. Replace duplicate ssoadapter meta-template names with unique values.

      For example, if there are two `SUN-UWC-MAIL` ssoadapter meta-templates, replace the `configName` value for one of them with `SUN-UWC-MAIL2`, resulting in two uniquely named templates:

      ```
      default|imap:/?configName=SUN-UWC-MAIL ...
      default|imap:/?configName=SUN-UWC-MAIL2 ...
      ```

3. Create an `amadmin` request file that will import the modified ssoadapter meta-templates, overwriting the original data.

   a. Copy `/tmp/ssoadapter-templates.xml` to `/tmp/ssoadapter-new-templates.xml`

   b. In /tmp/ssoadapter-new-templates.xml, replace the string:

      ```
      sunConfigurationTemplates=[
      ```

      with:

      ```
      <?xml version="1.0" encoding="ISO-8859-1"?>
      <!DOCTYPE Requests
        PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
        "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
      >
      <Requests>
        <SchemaRequests serviceName="SunSSOAdapterService"
        SchemaType="Global">
          <ModifyDefaultValues>
            <AttributeValuePair>
              <Attribute name="sunConfigurationTemplates"/>
      ```

c. Replace all ampersands ("&") with "&amp;".

For example, the line:

```
default|imap:/?configName=SUN-UWC-MAIL
&proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&subType=sun-one&enableProxyAuth=false...
```

would become:

```
default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...
```

d. Remove the commas (",") at the end of each ssoadapter meta-template.

e. Wrap each ssoadapter meta-template with a beginning <Value> tag and ending </Value> tag.

For example:

```
<Value>default|imap:/?configName=SUN-UWC-MAIL
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...</Value>
```

f. Remove the closing bracket ("]") from the last ssoadapter meta-template.

g. Add the following lines at the end of the file:

```
      </AttributeValuePair>
    </ModifyDefaultValues>
  </SchemaRequests>
</Requests>
```

The resulting ssoadapter-new-templates.xml file for the single template used in the above steps should look like the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Requests
  PUBLIC "-//iPlanet//iDSAME 5.0 Admin CLI DTD//EN"
  "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<Requests>
  <SchemaRequests serviceName="SunSSOAdapterService"
  SchemaType="Global">
    <ModifyDefaultValues>
      <AttributeValuePair>
        <Attribute name="sunConfigurationTemplates"/>
<Value>default|imap:/?configName=SUN-UWC-MAIL
```

```
&amp;proxyAdminPassword=%5BPROXY-ADMIN_PASSWORD%5D
&amp;subType=sun-one&amp;enableProxyAuth=false ...</Value>
      </AttributeValuePair>
    </ModifyDefaultValues>
  </SchemaRequests>
</Requests>
```

4. Import the new `ssoadapter-new-templates.xml` file.

   ```
   AccessManager-base/bin/amadmin -u amadminUser -w password -v
   -t ssoadapter-new-templates.xml
   ```

At this point, you can access the ssoadapter tab in the Portal Server Console to see the updated ssoadapters.

Enabling the URLScrapper Channel

When upgrading from Release 3 to Release 5, you have to enable the URLScrapper channel. See "Enable the URLScrapper Channel" on page 336.

Delete Gateway Service Entry

The `amService-srapGateway` user entry must be manually deleted when upgrading Portal Server from Release 2, otherwise the Portal Server Secure Remote Access Gateway component, if used, will fail to start after upgrade. Perform the following steps:

1. Log in to Access Manager Console.
2. List all users in the organization DN.
3. Delete the `amService-srapGateway` user.

Multiple Instance Upgrades

Multiple instance rolling upgrades (see"Multiple Instance Upgrades" on page 342) are not supported in upgrading Release 2 Portal Server to Release 5.

Upgrading Portal Server from the Interim Feature Release 7.0

This section includes information about upgrading Portal Server from the Interim Feature Release (IFR) 7.0 2005Q4 to Java ES 5 (Release 5).

The section covers the following topics:

- Portal Server IFR Upgrade Introduction
- Portal Server IFR 7.0 Upgrade
- Multiple Instance Upgrades

Portal Server IFR Upgrade Introduction

When upgrading Portal Server IFR 7.0 to Release 5 Portal Server, consider the following aspects of the upgrade process:

- The Portal Server IFR is not supported on Application Server 7.*x*, hence Upgrade Scenario 5 in Table 15-4 on page 316 does not apply.
- The `psupgrade` script, used for upgrading Portal Server IFR to Release 5, does not install new packages, as in the case of upgrade from Release 4. Instead, the upgrade procedure will require you to apply the following patches:

Table 15-8 Patches[1] to Upgrade Portal Server IFR to Release 5

Description	Patch ID: Solaris 9 & 10	Patch ID: Linux
Portal Server 7.1	121465-28 (SPARC) 121466-28 (x86)	121467-28
Portal Server 7.1 localization	123254-02 (SPARC) 124590-02 (x86)	123255-02

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. If newer revisions become available, use the newer ones instead of those shown in the table.

- The psupgrade script does not support upgrade of Portal Server IFR to Release 5 for Release 5 Web Server. There are two implications of this limitation:
  - If Web Server has already been upgraded to Release 5, you cannot upgrade the Portal Server IFR software deployed in the earlier Web Server container. Web container upgrade Scenario 2 in Table 15-4 on page 316 is therefore not supported. Note the warning in "Special Cases" on page 57.
  - If Web Server has *not* already been upgraded to Release 5, you can first upgrade the Portal Server IFR software in the earlier Web Server (6.x) container to Release 5, as documented in "Portal Server IFR 7.0 Upgrade," below, and then subsequently upgrade Web Server (and, if necessary, Access Manager) to Release 5. If this is your upgrade scenario, also see "Upgrade in Release 5 Web Server (7.0) Web Container" on page 370 for additional post-upgrade instructions.

Portal Server IFR 7.0 Upgrade

This section describes how to perform an upgrade of Portal Server from the IFR to Java ES Release 5 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:

- IFR 7 Pre-Upgrade Tasks
- Upgrading Portal Server IFR 7.0 (Solaris)
- Upgrading IFR 7 Portal Server (Linux)
- Verifying the Upgrade
- IFR 7 Post-Upgrade Tasks
- Rolling Back the Upgrade (Solaris)
- Rolling Back the Upgrade (Linux)

IFR 7 Pre-Upgrade Tasks

Pre-upgrade tasks for the IFR upgrade are the same as for the Release 4 upgrade (see "Release 4 Pre-Upgrade Tasks" on page 319), with the following exceptions:

- Obtain Required Configuration Information
- Configuration of Common Agent Container

Obtain Required Configuration Information

The information required by the `psupgrade` script, as detailed in "Obtain Required Configuration Information and Passwords" on page 321, does not fully apply to upgrade from Portal Server IFR. Because Portal Server IFR is not supported on Application Server 7.*x*, web container upgrade Scenario 5 in Table 15-6 on page 321 is not applicable.

Configuration of Common Agent Container

Common Agent Container is a shared component that provides container services for Java ES monitoring and management agents. Portal Server administrative tools such as Portal Server Console and the `psadmin` command line interface use a set of monitoring and management agents, collectively called the Portal Administration Server, that are deployed in Common Agent Container.

If Java ES shared components have been upgraded to Release 5 before you perform the upgrade of Portal Server IFR 7.0, then the following additional steps are required for you to log in to Release 5 Portal Server Console and to use the `psadmin` command line interface. (If Java ES shared components have *not* been upgraded to Release 5 before you perform the upgrade of Portal Server IFR 7.0, then ignore the following additional steps.)

1. Reconfigure Common Agent Container.

   ```
   PortalServer7-base/bin/psconfig --config
        PortalServer7-base/samples/example2.xml
   ```

   The `example2.xml` file provides reconfiguration information. You must first edit the `example2.xml` file to provide necessary passwords before running the `psconfig` command. If you are using non-default Portal Server locations, you must also provide the correct directories.

2. Edit the web container's classpath to reference Common Agent Container.

   The web container's classpath will contain a reference to the location of the previous Common Agent Container release: (*rel4CAC-base-dir*`/lib/cacao_cacao.jar`).

   Replace this reference with the Release 5 location: (*rel5CAC-admin-dir*`/lib/cacao_cacao.jar`).

3. Check that the property file, `pasconnect.properties has been created` in the *PortalServer7Config-base* directory with the following property:

   ```
   pas.host=
   ```

   The property value can be null, `localhost`, or the actual Portal Server host name.

4. Restart Common Agent Container.

```
rel5CAC-admin-dir/bin/cacaoadm start
```

Upgrading Portal Server IFR 7.0 (Solaris)

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

IFR 7 Upgrade Considerations (Solaris)

The Portal Server IFR upgrade to Java ES Release 5 takes into account the same considerations as the Release 4 upgrade (see "Upgrade Considerations (Solaris)" on page 324).

In addition, see the issues raised in "Portal Server IFR Upgrade Introduction" on page 359.

IFR 7 Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

```
su -
```

2. Stop any instances of the Portal Server Secure Remote Access Gateway, Rewriter Proxy, or Netlet Proxy that might be running locally.

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
   -f passwordFile -t gateway  -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
   -f passwordFile -t rwproxy -N gatewayProfileName
```

```
PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
   -f passwordFile -t nlproxy -N gatewayProfileName
```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

3. Make sure Access Manager is running if it is deployed to a web container different from the one to which Portal Server is deployed.

4. If not already running, start Portal Server by starting the web container to which it is deployed.

   Web Server 6.*x:*

   ```
   WebServer-base/https-instanceName/start
   ```

   Application Server *8.x:*

   ```
   AppServer8-base/bin/asadmin start-domain --user admin_ID
         --password password domainName
   ```

5. Obtain the required patch, based on Table 15-8 on page 359.

   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

6. Apply the appropriate Portal Server patch and, if needed, localization patch in Table 15-8.

   ```
   patchadd /tmp/patch_ID
   ```

7. Confirm that the patch upgrades were successful:

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

   The output should return the versions of patch IDs applied in Step 6.

8. In cases where localization packages have been upgraded in Step 6, set the Portal Server Console JVM's locale to UTF-8.

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

9. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script:

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

10. Make sure you have adequate swap space on your computer.

    As a guideline, the swap space should be set to twice the amount of physical ram.

11. Run the psupgrade script.

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    The psupgrade script is not run from the Java ES Release 5 distribution and does not invoke the Java ES installer (the packages were already patched).

    The script queries the system to detect the location and port number and other information regarding the web container to which you are deploying Portal Server web applications. Depending on web container upgrade scenario (see Table 15-4 on page 316), the script requests you to input additional information required to deploy Portal Server to the appropriate web container.

    Table 15-6 on page 321 shows the information requested for the different web container upgrade scenarios in Table 15-4 on page 316.

NOTE Be sure you enter correct values for psupgrade parameters, as you can't go back and change them, and it is also very difficult to roll back changes made by the psupgrade script.

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.

Upgrading IFR 7 Portal Server (Linux)

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

IFR 7 Upgrade Considerations (Linux)

The upgrade of Portal Server IFR software to Release 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "IFR 7 Upgrade Considerations (Solaris)" on page 362), except that installing the Release 5 patches on Linux OS removes the previous RPMs.

IFR 7 Upgrade Procedure (Linux)

The procedure documented below applies to Portal Server on the computer where the upgrade is taking place.

CAUTION An upgrade from Portal Server IFR to Release 5 on Linux cannot be rolled back. Make sure you back up your system *before* performing the following procedure.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop any instances of the Portal Server Secure Remote Access Gateway, Rewriter Proxy, or Netlet Proxy that might be running locally.

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t gateway  -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t rwproxy -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t nlproxy -N gatewayProfileName
   ```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

3. Make sure Access Manager is running if it is deployed to a web container different from the one to which Portal Server is deployed.

4. If not already running, start Portal Server by starting the web container to which it is deployed.

   Web Server 6.*x:*
   WebServer-base`/https-`*instanceName*`/start`

   Application Server *8.x:*

   ```
   AppServer8-base/bin/asadmin start-domain --user admin_ID
        --password password domainName
   ```

5. Obtain the required patch using the patch numbers and RPM names from Table 15-8 on page 359.

   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`

6. Apply the Portal Server patch and, if needed, localization RPMs for Portal Server in Table 15-8 on page 359, in that order.

   See the Readme file for the Portal Server patch, which describes how to use a script to apply the patch's RPMs:

   ```
   cd /tmp
   ```

   where /tmp is the directory to which you download the patch.

   ```
   ./upgradeportalrpm
   ```

   The update script installs the RPM's.

   For the localization patch, install each RPM using the following command:

   ```
   rpm -Fvh patchName-version.rpm
   ```

7. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal
   ```

   The upgrade revision numbers of the RPMs should be returned.

8. In cases where localization packages have been upgraded in Step 6, set the Portal Server Console JVM's locale to UTF-8.

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

9. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script:

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

10. Make sure you have adequate swap space on your computer.

    As a guideline, the swap space should be set to twice the amount of physical ram.

11. Run the `psupgrade` script.

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    The `psupgrade` script is not run from the Java ES Release 5 distribution and does not invoke the Java ES installer (the packages were already patched).

The script queries the system to detect the location and port number and other information regarding the web container to which you are deploying Portal Server web applications. Depending on web container upgrade scenario (see Table 15-4 on page 316), the script requests you to input additional information required to deploy Portal Server to the appropriate web container.

Table 15-6 on page 321 shows the information requested for the different web container upgrade scenarios in Table 15-4 on page 316.

NOTE Be sure you enter correct values for `psupgrade` parameters, as you can't go back and change them, and it is also very difficult to roll back changes made by the `psupgrade` script.

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.

Verifying the Upgrade

You can verify the patching of Portal Server packages to Release 5 using the following command:

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile.
```

See Table 15-5 on page 319 for output values.

To verify the full upgrade, confirm that the Portal Desktop comes up and the `psadmin` administration utility functions as documented.

You can also check the following upgrade log files at `/var/sadm/install/logs`:

- `Sun_Java_System_Portal_Server_upgrade.log`
- `Sun_Java_System_Portal_Server_upgrade.log_ant_`*xxx*`.log`

  where *xxx* can be `preupgrade`, `upgrade`, or `postupgrade`.

IFR 7 Post-Upgrade Tasks

When upgrading Portal Server from IFR 7 to Release 5, you must perform the post-upgrade procedures required to address the following situations:

- Enabling the Java ES Monitoring Framework
- Upgrade in Application Server Web Container
- Upgrade in Release 5 Web Server (7.0) Web Container
- Redeploy Custom Portlet Applications
- Migrate Customized Portlet Applications
- Correct Links in Bookmark and Application Channels
- Manual Migration of Struts-based Portlets

Enabling the Java ES Monitoring Framework

Enabling the Java ES Monitoring Framework (MFWK shared component) lets Portal Server administrative tools, such as Portal Server Console and the `psadmin` command line interface, report metrics like the number of visitors and the portals they frequent. To enable MFWK:

1. Locate the following two files:

   MFWK-base`/template/jesmf/desktopmfwk.properties`
   MFWK-base`/template/jesmf/com.sun.cmm.ps.xml`

   where *MFWK-base* is the following path:

   `/opt/SUNWmfwk` (Solaris)

   `/opt/sun/mfwk` (Linux)

2. Copy the two files to the following directory:

   PortalServer7Data-base`/portals/portal_ID/config/Portal_Instance/`

3. In the `desktopmfwk.properties` file, replace

   ```
   com.sun.portal.ProductCollectionId=%PS_DIR%
   ```

   with

   ```
   com.sun.portal.ProductCollectionId=Portal_Installed_Location
   ```

4. Add the following two jar files:

   ```
   MFWK-base/lib/mfwk_instrum_tk.jar
   MFWK-base/lib/mfwk_agent.jar
   ```

   to the appropriate web container classpath (`server.xml` file for Web Server and `domain.xml` for Application Server)

5. Restart the corresponding web container.

Upgrade in Application Server Web Container

If Portal Server is deployed in an Application Server web container, then you must perform the following additional procedure to successfully redeploy Portal Server:

1. Find the section that defines `psconsole` settings in the *AppServer8Config-base*`/domains/`*domainName*`/config/server.policy` file:

2. Add the following line at the end of that section:

   ```
   permission java.lang.RuntimePermission "getProtectionDomain"
   ```

3. Restart the Application Server instance.

   ```
   AppServer8-base/bin/asadmin start-domain --user admin_ID
        --password passworddomainName
   ```

   ```
   AppServer8-base/bin/asadmin start-node-agent --user admin_ID
        --password password nodeagentName
   ```

   where *nodeagentName* has the form *hostName_domainName*, but is simply *hostName* by default.

Connections Hang in Application Server Web Container

When the upgraded Portal Server is deployed in an Application Server web container, portal applications can hang waiting to get Java DB connections. To address this problem, perform the following steps:

1. Remove settings in *PortalServer7Data-base*`/derby/derby.properties` for the following 2 parameters:

   ```
   derby.drda.maxThreads
   derby.drda.timeslice
   ```

2. Restart Java DB.

   ```
   ANT_HOME/bin/ant
       -DPS_CONFIG=PortalServer7Config-base/PSConfig.properties
       -buildfile PortalServer7-base/lib/derby.xml
       [stop-instance|start-instance]
   ```

   where ANT_HOME is /usr/sfw (on Solaris) and /opt/sun (on Linux).

3. Change Java DB configuration settings for Application Server.

   Using the Application Server Console, change attribute values for the following connection pool resources: communitymcPool, FileSharingDBPool, PointBasePool, SurveyDBPool.

   Change the following attribute values as follows:

   ```
   Idle Timeout to 300 or more
   Resource Type to javax.sql.ConnectionPoolDataSource
   Datasource classname to
       org.apache.derby.jdbc.ClientConnectionPoolDataSource
   ```

4. Restart the Application Server instance in which Portal Server is deployed.

Upgrade in Release 5 Web Server (7.0) Web Container

In the case where you have upgraded Portal Server IFR in a Web Server that has *not* already been upgraded to Release 5 because the psupgrade script does not support upgrade of Portal Server IFR on Release 5 Web Server (see "Portal Server IFR Upgrade Introduction" on page 359), you must perform the following additional post-upgrade steps:

1. Upgrade Web Server (and, if necessary, Access Manager) to Release 5.

2. Reconfigure Web Server container values needed by Portal Server Console and the psadmin command line interface.

   a. Open an LDAP browser.

      Configuration values are stored in Directory Server.

   b. Under DN, look for:

      ```
      sunPortalAdminPortalDomainID=defaultDomain
      ->sunPortalAdminPortalDomainPortalID=portal1
      ->sunPortalAdminPortalDomainPortalServerInstanceIn=host-port
      ```

c. Perform edits as follows.

   Note that all entries are prefaced by the following string: `sunPortalAdminPortalDomainPortalServerInstance`

   - Delete the entry for `WebContainerInstanceDir`.
   - Add an entry for `WebContainerDomainName` and assign it the value of `Web Container Config Name` from Table 15-6 on page 321.
   - Edit entries such as `InstallDir`, `WebContainerType`, `DocRoot`, and other parameters shown in Table 15-6 on page 321 to correspond to Release 5 Web Server (7.0) values.

3. Create a Release 5 Portal Server instance.

   ```
   PortalServer7-base/bin/psadmin create-instance newInstance_ID
   ```

   If the value of *newInstance_ID* already exits, an error will be thrown, so it is advantageous to perform this step before Step 4 below.

4. Delete the Portal Server IFR instance.

   ```
   PortalServer7-base/bin/psadmin delete-instance oldInstance_ID
   ```

Redeploy Custom Portlet Applications

If you have created and deployed custom portlet applications, then these portlets must be manually redeployed after upgrade to Release 5 Portal Server. Even though display profile entries will exist and the channel name will be displayed, content will not be seen until you redeploy your custom portlets.

Redeploy portlets using the following command:

```
PortalServer7-base/bin/psadmin deploy-portlet
```

You can confirm redeployment by looking for the corresponding `.war` and XML files in the following location:

```
PortalServer7Data-base/portals/Upgraded/war
```

Migrate Customized Portlet Applications

Portlet applications based on the user interface framework provided by Sun Java Web Console (SJWC) need to be manually migrated to Release 5 and redeployed.

In particular, this requirement applies to four web applications distributed with Portal Server as sample portlet applications meant to be customized and installed into a portal: `filesharing`, `surveys`, `wiki`, `rssportlet`. During upgrade, bug-fixed versions of these sample portlet applications are placed on disk in the portlet applications area. If you have customized these portlet applications for your own use, you need to manually migrate them to Release 5 and redeploy them; they are not handled by the Portal Server upgrade process.

By default, some of these portlet applications (`filesharing` and `surveys`) are deployed and available for users when a community is created.

You can upgrade, customize, and redeploy SJWC-based portlet applications using the following procedure. The `filesharing` portlet application is used as an example:

1. **Extract the Release 5 SJWC jar files.**

   a. `mkdir /tmp/lh`

   b. `cd /tmp/lh`

   c.
   ```
   /usr/jdk/entsys-j2se/bin/jar xvf
       PortalServer7-base/portlet/communityportlets.war
       WEB-INF/lib/commons-beanutils.jar
       WEB-INF/lib/commons-collections-3.1.jar
       WEB-INF/lib/commons-digester.jar
       WEB-INF/lib/commons-logging.jar
       WEB-INF/lib/dataprovider.jar WEB-INF/lib/jsf-api.jar
       WEB-INF/lib/jsf-impl.jar WEB-INF/lib/webui.jar
   ```

   If *PortalServer7-base*`/portlet/communityportlets.war` is not found, use *PortalServer7-base*`/portlet/core/communityportlets.war`.

   d. **Rename one of the files.**

   ```
   mv WEB-INF/lib/commons-collections-3.1.jar
   WEB-INF/lib/commons-collections.jar
   ```

2. **Locate the `filesharing` portlet application.**

   ```
   cd PortalServer7Config-base/portals/portal1/portletapps/filesharing
   ```

3. **Inject the updated SJWC libraries.**

   ```
   jar uvf src/filesharing.war.tokenized -C /tmp/lh WEB-INF
   ```

4. **Customize the `filesharing` portlet application.**

   ```
   ant customize
   ```

5. Redeploy the `filesharing` portlet application.

   a. *PortalServer7-base*`/bin/psadmin undeploy-portlet -u` *amadmin* `-f passwordfile -p` *portal_id* `-i` *instance_id* `-g filesharing`

   b. `ant deploy`

   c. Go to the following directory (depending on web container):

      Application Server 8.x:

      AppServer8Config-base`/domains/domain1/applications/j2ee-modules/communityportlets/WEB-INF`

      Web Server 6.x:

      WebServer6-base`/https-`*instanceName*`/webapps/https-`*instanceName*`/communityportlets/WEB-INF`

      Web Server 7.x:

      WebServer7Config-base`/https-`*configName*`/web-app/https-`*configName*`/communityportlets/WEB-INF`

   d. Open the `sun-web.xml` file and add the following line just before the last line (that is, before the `sun-web-app` end tag):

      ```
      <class-loader delegate="false"/>
      ```

   e. Repeat Step c and Step d for the `sun-web.xml` file under `filesharing/WEB-INF`.

6. Repeat Step 2 through Step 5 for `surveys` and any other custom portlet application based on the SJWC framework.

7. Restart the web container.

Correct Links in Bookmark and Application Channels

When upgrading Release 4 Portal Server to Release 5, the bookmark and application channels have duplicate and spurious links. To fix these links, perform the following procedure.

1. Log in to PSConsole.

2. From the Common Tasks tab, click on Manage Channels & Containers.

3. Choose DeveloperSample [Org] as DN and click OK.

4. Select JSPTabContainer [default] as the View Type.

5. Under MyFrontPageTabPanelContainer, click on App Channel.

   App channel properties will be displayed on the right-hand side frame.

   To view properties for a specific locale, Click on Table Preferences and provide the Locale value: `de`, `fr`, `es`, `ja`, `ko`, `zh`, `zh_CN`, `zh_TW`.

6. Edit the userApps property.

   a. In userApps property click on [Edit Values...] link

      A pop-up window with existing applications will appear.

   b. Add following application to the list:

      NetFile

   c. Click on Save and then Close.

7. Edit the target property.

   a. In target property click on [Edit Values...] link

      A pop-up window with existing targets will appear.

   b. Add the following target to the list:

      ```
      NetFile|/portal/NetFileApplet?Refer=java2
      ```

   c. Click on Save and then Close.

Manual Migration of Struts-based Portlets

If you have custom portlets that use code based on the Struts framework, then these portlets need to be manually updated to use the `struts.jar` file included in Release 5 Portal Server. Use the following procedure:

1. Undeploy the struts-based portlet application.

   PortalServer7-base`/bin psadmin undeploy-portlet`

2. Update the `.war` file with the correct version of the `struts.jar` file.

   Copy *PortalServer7-base*`/lib/struts.jar` to *strutsbasedPortlet*`/WEB-INF/lib/struts.jar`

   where *strutsbasedPortlet* is the directory in which the struts-based portlet files reside.

3. Create a `.war` archive of the *strutsbasedPortlet* directory.

4. Redeploy the portlet application.

   PortalServer7-base`/bin psadmin deploy-portlet`

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 5 consists of reverting back to the IFR installation at *PortalServer7-base* and redeploying the IFR web applications.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Restore Directory Server to the state it was in before upgrade.

   Use the Directory Server backup/restore command line and GUI utilities. See the Directory Server Backup and Restore chapter of the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*, http://docs.sun.com/doc/819-0995.

3. Undeploy the Release 5 Portal Server web applications that were re-deployed during the upgrade to Release 5.

   Use the web container's administration utilities (command line or console) to undeploy the following packages:

   ```
   portal
   psconsole
   search1
   wsssoportlet
   guessnumber
   portletsamples
   ```

4. Stop Portal Server by stopping its web container.

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/stop

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/stopserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/stopserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin stop-domain --user *admin_ID*
   --password *password domainName*

5. Back out the Portal Server 7.1 patch in Table 15-8.

   patchrm *patch_ID*

6. Restart Portal Server by restarting its web container.

   Web Server 6.*x:*
   WebServer-base/https-*instanceName*/start

   Web Server 7.0*:*
   Admin Server--
   WebServer7Config-base/admin-server/bin/startserv
   Instance Server--
   WebServer7Config-base/https-*configName*/bin/startserv

   Application Server *8.x:*
   AppServer8-base/bin/asadmin start-domain --user *admin_ID*
   --password *password domainName*

7. Deploy the Release 5 Portal Server web applications that were un-deployed during Step 3 on page 375.

   Use the web container's administration utilities (command line or console) to deploy the packages.

8. 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.

Rolling Back the Upgrade (Linux)

Rollback of the upgrade cannot be performed on Linux.

However, you can create a parallel system *before* upgrading and testing that system before attempting an upgrade. If you need to roll back the upgrade, you can revert back to that parallel system.

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. To perform a rolling upgrade from the IFR to Release 5, use the same procedure documented in "Multiple Instance Upgrades" on page 342, except substitute the IFR wherever Release 4 is referenced.

Chapter 16

Portal Server Secure Remote Access

This chapter describes how to upgrade Portal Server Secure Remote Access to Java ES 5 (Release 5): Sun Java System Portal Server Secure Remote Access 7.1.

The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Release 5. The chapter covers upgrades on both the Solaris and Linux operating systems:

- "Overview of Portal Server Secure Remote Access Upgrades" on page 381
- "Upgrading Portal Server Secure Remote Access from Java ES Release 4" on page 386
- "Upgrading Portal Server Secure Remote Access from Java ES Release 3" on page 405
- "Upgrading Portal Server Secure Remote Access from Java ES Release 2" on page 407
- "Upgrading Portal Server Secure Remote Access from the Interim Feature Release 7.0" on page 410

NOTE File locations in this chapter are specified with respect to directory paths referred to as *PortalServer6-base and PortalServer6Config-base* (Portal Server 6.*x*) and *PortalServer7-base and PortalServer7Config-base* (Portal Server 7.*x*). 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 values of these directory paths are shown in the following table.

Table 16-1 Portal Server Secure Remote Access Directory Paths

Path Name Variable	Solaris OS	Linux OS
PortalServer6-base	`/opt/SUNWps`	`/opt/sun/portal`
PortalServer6Config-base	`/etc/opt/SUNWps`	`/etc/opt/sun/portal`
PortalServer6Data-base	`/var/opt/SUNWps`	`/var/opt/sun/portal`
ortalServer7-base	`/opt/SUNWportal`	`/opt/sun/portal`
PortalServer7Config-base	`/etc/opt/SUNWportal`	`/etc/opt/sun/portal`
PortalServer7Data-base	`/var/opt/SUNWportal`	`/var/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 5 (Release 5):

- About Java ES Release 5 Portal Server Secure Remote Access
- Portal Server Secure Remote Access Upgrade Roadmap
- Portal Server Secure Remote Access Data
- Portal Server Secure Remote Access Upgrade Strategy

About Java ES Release 5 Portal Server Secure Remote Access

Portal Server Secure Remote Access (consisting of Gateway, Rewriter Proxy, Netlet Proxy components is closely coupled to Portal Server, though usually deployed on computers different from the one hosting Portal Server. Portal Server Secure Remote Access components use the same administrative infrastructure as Portal Server proper and interact with servlets and applets residing on the computer hosting Portal Server.

Java ES Release 5 Portal Server Secure Remote Access represents a major release with respect to Release 4, with many new enhancements and features. Many of these changes were made in an Interim Feature Release (IFR) subsequent to Release 4. Release 5 represents only minor feature changes with respect to the IFR. For information about the IFR enhancements and new features, see the *Sun Java System Portal Server 7.1 Release Notes*, `http://docs.sun.com/doc/819-4986/6n4l3f365?a=view.` In particular, the Release 4 command line administrative interface has been replaced by the `psadmin` command.

Portal Server Secure Remote Access Upgrade Roadmap

Table 16-2 shows the supported Portal Server Secure Remote Access upgrade paths to Java ES Release 5. The table applies to both Solaris and Linux operating systems.

Table 16-2 Upgrade Paths to Java ES 5 (Release 5): Portal Server Secure Remote Access 7.1

Java ES Release	Portal Server Secure Remote Access Version	General Approach	Reconfiguration Required
Interim Feature Release (IFR)	Sun Java System Portal Server Secure Remote Access IFR 7.0 2005Q4	Direct upgrade: Performed by applying patches and then using an upgrade script.	None.
Release 4	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q4	Direct upgrade: Performed using an upgrade script.	None.
Release 3	Sun Java System Portal Server Secure Remote Access 6.3.1 2005Q1	Direct upgrade: Performed using an upgrade script.	None.
Release 2	Sun Java System Portal Server Secure Remote Access 6.3 2004Q2	Direct upgrade: Performed using an upgrade script.	None.
Release 1	Sun ONE Portal Server Secure Remote Access 6.2 (2003Q4)	No direct upgrade: But can be performed by upgrading first to Release 3 and then upgrading from Release 3 to Release 5.	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 16-3 Portal Server Secure Remote Access Data Usage

Type of Data	Location	Usage
Configuration data	*PortalServer6Config-base/*	Configuration of Portal Server Secure Remote Access.
Directory schema Services configuration User data	Directory Server	Portal Server Secure Remote Access depends on services configurations, such as the portal desktop, and 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.

Portal Server Secure Remote Access Upgrade Strategy

Your strategy for upgrading Portal Server Secure Remote Access generally depends on the many considerations discussed in Chapter 1, "Planning for Upgrades": upgrade path, dependencies between Java ES components, selective upgrade versus upgrade all, multi-instance deployments, and so forth.

This section is to particularize that general discussion to Portal Server Secure Remote Access by presenting issues that might influence your Portal Server Secure Remote Access upgrade plan.

Compatibility Issues

Release 5 Portal Server Secure Remote Access introduces public interface changes in the `psadmin` command used to start and stop Gateway, Rewriter Proxy, and Netlet Proxy components. See the *Sun Java System Portal Server 7.1 Command-Line Reference*, `http://docs.sun.com/doc/819-5030`.

Individual Portal Server Secure Remote Access components (including the Gateway, the Rewriter Proxy, and the Netlet Proxy) are not backwardly compatible with earlier versions; all need to be synchronized, along with Portal Server itself, at Java ES Release 5. This requirement applies to Portal Server Secure Remote Access components that are local as well as distributed.

In addition, there is an incompatibility between the Directory Server data structures used by Release 5 Portal Server and earlier Portal Server versions. This incompatibility impacts a rolling upgrade of multiple Portal Server instances using the same Directory Server data.

Portal Server Secure Remote Access Dependencies

Portal Server Secure Remote Access is closely coupled with Portal Server, depending on software packaged with Portal Server and running on the same computer as Portal Server.

However, Portal Server Secure Remote Access also depends on other Java ES components. These dependencies can impact your 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 components have dependencies on the following Java ES components:

- **Shared components.** Portal Server Secure Remote Access components have dependencies on specific Java ES shared components (see Table 1-9 on page 47).
- **Portal Server** Portal Server Secure Remote Access components have a mandatory dependency on Portal Server, which includes local components that are needed to support Portal Server Secure Remote Access functions.
- **Access Manager (or Access Manager SDK).** Portal Server Secure Remote Access components have a mandatory dependency on 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 has a mandatory dependency on Directory Server, which stores user data. As a result, Portal Server Secure Remote Access upgrades might require extensions of directory schema.

Selective Upgrade Issues

While, in general, Java ES Release 5 supports selective upgrade of all components on a computer, the fact that Portal Server Secure Remote Access is closely tied to Portal Server means that Portal Server Secure Remote Access must be upgraded if Portal Server is upgraded. Similarly, upgrade of Portal Server Secure Remote Access requires that Portal Server also be upgraded.

As a result, the upgrade of Portal Server Secure Remote Access is bound by the same restrictions as Portal Server (see Portal Server "Selective Upgrade Issues" on page 315): you can either upgrade Portal Server Secure Remote Access and *all* of its product component dependencies to Release 5, or upgrade only Portal Server Secure Remote Access and Portal Server to Release 5, leaving other product component dependencies at Release 4.

Dual Upgrade

Dual upgrades, in which both Portal Server Secure Remote Access and operating system are upgraded (as described in "Dual Upgrades: Java ES and Operating System Softwared" on page 43) can be performed using the in-place operating system upgrade approach:

1. Back up existing Portal Server Secure Remote Access data.

   See "Portal Server Secure Remote Access Data" on page 383 for the location of essential data.

2. Upgrade the operating system.

   The upgrade leaves the existing file system in place.

3. Upgrade to Release 5 Portal Server Secure Remote Access.

   See the appropriate section of this chapter, depending on upgrade path.

Upgrading Portal Server Secure Remote Access from Java ES Release 4

This section includes information about upgrading Portal Server Secure Remote Access from Java ES 2005Q4 (Release 4) to Java ES 5 (Release 5).

The section covers the following topics:

- Introduction
- Release 4 Portal Server Secure Remote Access Upgrade
- Multiple Instance Upgrades

Introduction

When upgrading Java ES Release 4 Portal Server Secure Remote Access to Release 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed using an upgrade script, `psupgrade`. The script removes old packages, installs new packages, and migrates configuration data when necessary.
- **Upgrade Dependencies.** Portal Server Secure Remote Access has dependencies on a number of Java ES shared components (see Table 1-9 on page 47). While Release 5 Portal Server Secure Remote Access is compatible with the Release 4 version of these shared components, upgrade of shared components is nevertheless necessary because the `psupgrade` script used to upgrade Portal Server Secure Remote Access requires the Release 5 version of the ANT shared component.

  Release 5 Portal Server Secure Remote Access also has dependencies upon Portal Server, Access Manager, and Directory Server, as described in "Portal Server Secure Remote Access Dependencies" on page 384. Two approaches to upgrading these dependencies are supported (see "Selective Upgrade Issues" on page 385):
- **Upgrade Dependencies.** Portal Server Secure Remote Access has dependencies on a number of Java ES shared components (see Table 1-9 on page 47), however Release 5 Portal Server Secure Remote Access is compatible with the Release 4 version of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Portal Server Secure Remote Access to Release 5.

However, Release 5 Portal Server Secure Remote Access has a hard upgrade dependency only on Portal Server. Release 5 Portal Server Secure Remote Access also has soft upgrade dependencies upon Access Manager and Directory Server, as described in "Portal Server Secure Remote Access Dependencies" on page 384.

Two approaches to upgrading these product component dependencies are supported (see "Selective Upgrade Issues" on page 385):

- All dependencies satisfied by Release 4 components (*none* except Portal Server are upgraded to Release 5)
- All dependencies satisfied by Release 5 components (*all* are upgraded to Release 5).

The approach taken for Portal Server Secure Remote Access must be the same as the approach taken by Portal Server.

- **Backward Compatibility.** Release 5 Portal Server Secure Remote Access is backwardly compatible with the Release 4 version.
- **Upgrade Rollback.** Rollback of the Release 5 upgrade of Portal Server Secure Remote Access to Release 4 consists of restoring Release 4 packages and restoring Release 4 Directory data.
- **Platform Issues.** The general approach for upgrading Portal Server Secure Remote Access is the same on both Solaris and Linux operating systems, however release 5 Portal Server Secure Remote Access is installed in a new path on Solaris OS, but in the same Release 4 path on Linux OS.

Release 4 Portal Server Secure Remote Access Upgrade

This section describes how to perform an upgrade of Portal Server Secure Remote Access from Java ES Release 4 to Java ES Release 5 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:

- Release 4 Pre-Upgrade Tasks
- Upgrading Release 4 Portal Server Secure Remote Access (Solaris)
- Upgrading Release 4 Portal Server Secure Remote Access (Linux)
- Verifying the Upgrade

- Release 4 Post-Upgrade Tasks
- Rolling Back the Upgrade (Solaris)
- Rolling Back the Upgrade (Linux)

Release 4 Pre-Upgrade Tasks

Before you upgrade Portal Server Secure Remote Access you should perform the following tasks:

- Verify Current Version Information
- Upgrade Portal Server Secure Remote Access Dependencies
- Back Up Release 4 Portal Server Secure Remote Access Configuration Information
- Remove Configuration for Load Balancer
- Remove Configuration for Directory Proxy Server
- Obtain Required Configuration Information and Passwords

Verify Current Version Information

You can verify the current version of Portal Server Secure Remote Access using the following command:

```
PortalServer6-base/bin/version
```

Table 16-4 Portal Server Secure Remote Access Version Verification Outputs

Java ES Release	Portal Server Secure Remote Access Version Number
Release 2	6.3
Release 3	6.3.1
Release 4	6.3.1[1]
IFR Release	7.0
Release 5	7.1

[1] The only difference between Release 3 and Release 4 is a patch. You can check for the Release 4 patches using the Solaris `showrev -p | grep` *patch_ID* command and the Linux `rpm -qa sun-portal-core` command and comparing the versions to those listed in the Java ES Release 4 *Upgrade Guide*.

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 5.

While Release 5 Portal Server Secure Remote Access is compatible with the Release 4 version of Java ES shared components, upgrade of shared components is nevertheless necessary because the `psupgrade` script used to upgrade Portal Server Secure Remote Access requires the Release 5 version of the ANT shared component.

In addition, Portal Server Secure Remote Access requires the upgrade of Portal Server. However it does not require upgrading other Java ES Release 4 product components upon which it depends.

In fact, your dependency upgrade approach is the same as that taken for Portal Server: if any of the dependencies are to be upgraded to Release 5, they all need to be upgraded (see "Selective Upgrade Issues" on page 315). However, because of the Portal Server Secure Remote Access dependency on Portal Server, the upgrade of Portal Server takes care of Portal Server Secure Remote Access dependencies, except, for shared components.

When you upgrade Portal Server Secure Remote Access dependencies to Release 5, the dependencies should be upgraded in the order below (skipping any that might already have been upgraded), before you upgrade Portal Server Secure Remote Access.

1. **Shared Components.** Instructions for synchronizing Java ES shared components to Release 5 are provided in "Upgrading Java ES Shared Components" on page 63.

2. **Portal Server.** Instructions for upgrading Portal Server are provided in Chapter 15, "Portal Server" on page 309.

Back Up Release 4 Portal Server Secure Remote Access Configuration Information

Upgrade of Portal Server Secure Remote Access to Release 5 does not require the reconfiguration of Portal Server Secure Remote Access software. However, as a safety measure the `psupgrade` script will back up the following directories where configuration information is stored:

PortalServer6Config-base/

Remove Configuration for Load Balancer

In cases in which Portal Server Secure Remote Access instances are accessed through a load balancer, the value of the `LOAD_BALANCER_URL` property used to configure such access can interfere with Portal Server Secure Remote Access upgrade. This setting must therefore be modified before performing upgrade of any Portal Server Secure Remote Access components. To modify the `LOAD_BALANCER_URL` property setting:

1. Note which of the following configuration files are locally resident (some of which support Portal Server components that might be locally installed):

   PortalServer6Config-base/`PSConfig.properties` (if Portal Server is local)
   PortalServer6Config-base/`GWConfig.properties` (if Gateway is local)
   PortalServer6Config-base/`RWPConfig.properties` (if Rewriter Proxy is local)
   PortalServer6Config-base/`NLPConfig.properties` (if Netlet Proxy is local)

2. Record the current value of the `LOAD_BALANCER_URL` property in these configuration files.

3. Modify the value of the `LOAD_BALANCER_URL` property to point to the corresponding Portal Server Secure Remote Access instance being upgraded:

   `LOAD_BALANCER_URL=`*hostName*`:`*port*`/portal`

4. Make sure that the following configuration properties, if present, reference the relevant Portal Server Secure Remote Access component (and *not* the load balancer), as shown below:

   In *PortalServer6Config-base*/`platform.conf.default` file:

   `gateway.host=`*Gateway_hostName*

   In *PortalServer6Config-base*/`GWConfig.properties` and *PortalServer6Config-base*/`GWConfig-default.properties` files:

   `GW_HOST=`*Gateway_hostName*
   `GW_IP=`*Gateway_hostIP*

   In *PortalServer6Config-base*/`RWPConfig.properties` and *PortalServer6Config-base*/`RWPConfig-default.properties` files:

   `RWP_HOST=`*RewriterProxy_hostName*
   `RWP_IP=`*RewriterProxy_hostIP*

   In *PortalServer6Config-base*/`NLPConfig.properties` and *PortalServer6Config-base*/`NLPConfig-default.properties` files:

   `NLP_HOST=`*NetLetProxy_hostName*
   `NLP_IP=`*NetLetProxy_hostIP*

Remove Configuration for Directory Proxy Server

In cases in which Portal Server Secure Remote Access instances access Directory Server through a Directory Proxy Server instance, the Directory Proxy Server host and port number settings must be modified before performing the upgrade and then restored to their original values after upgrade is complete.

To modify the appropriate settings:

1. Record the current value of the `DS_HOST` and `DS_PORT` properties in the following Access Manager configuration file:

   AccessManagerConfig-base`/config/AMConfig.properties`

2. Modify the value of the `DS_HOST` and `DS_PORT` properties to point directly to the relevant Directory Server instance.

Obtain Required Configuration Information and Passwords

Depending on the upgrade scenario, the `psupgrade` script requires you to input information about the following admin accounts:

- Directory Server Admin ID and password
- Access Manager Admin ID and password
- Directory Server amldapuser ID and password

Upgrading Release 4 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 Release 5 takes into account the following considerations:

- Portal Server Secure Remote Access software consists of subcomponents that perform a number of different roles, but must all be upgraded to Release 5 together:
  - **Portal-base. I**ncludes administrative Mbeans and accompanying administrative software, Logging Framework, and monitoring-related software, all of which are packaged into the SUNWportal-base package.
  - **Secure Remote Access applications.** Include the Gateway, Rewriter Proxy, and Netlet Proxy. These applications are normally deployed on one or more computers different from the computer hosting Portal Server proper. Secure Remote Access applications do not require a web container.

- When the Gateway, Rewriter Proxy and Netlet Proxy are not deployed on the same computer, then the Rewriter Proxy and Netlet Proxy should be upgraded before the Gateway is upgraded.
- All Portal Server Secure Remote Access subcomponents correspond to the same installed Portal Server Secure Remote Access image and, if present on the computer being upgraded, are upgraded at the same time.
- The `psupgrade` script automatically detects which Portal Server Secure Remote Access subcomponents are installed on the host computer and upgrades those components.

Upgrade Procedure (Solaris)

The procedure documented below applies to t he Portal Server Secure Remote Access component 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, synchronize all shared components to Release 5.

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

   This step is a necessary prerequisite to running the `psupgrade` script in Step 8 on page 393.

3. Stop any instances of the Gateway, Rewriter Proxy, or Netlet Proxy that are running locally.

   ```
   PortalServer6-base/bin gateway stop
   PortalServer6-base/bin netletd stop
   PortalServer6-base/bin rwproxyd stop
   ```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

4. Make sure Access Manager is running.

5. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script. For example,

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

6. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

7. If the Portal Server Secure Remote Access component you are upgrading is remote from Portal Server, copy the dpadmin executable from the computer hosting Portal Server to the computer hosting the Portal Server Secure Remote Access component.

   The dpadmin executable can be found in the following location:

   PortalServer7-base/SUNWps.bak/bin/dpadmin, if Portal Server has been upgraded.

   PortalServer6-base/bin/dpadmin, if Portal Server has not yet been upgraded.

8. Run the psupgrade script from the Java ES Release 5 distribution.

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   where *os_arch* matches your platform, such as Solaris_sparc.

   The psupgrade script invokes the Java ES installer to install new packages and requests the following information:

   - Directory Server Admin ID and password
   - Access Manager Admin ID and password
   - Directory Server amldapuser ID and password

9. Start instances of the Gateway, Rewriter Proxy, or Netlet Proxy that were stopped in Step 3 on page 392.

   ```
   PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
       -f passwordFile --name default --type gateway
   ```

   ```
   PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
       -f passwordFile --name default --type nlproxy
   ```

   ```
   PortalServer7-base/bin/psadmin start-sra-instance -u amadminUser
       -f passwordFile --name default --type rwproxy
   ```

   If the above commands fail, you must first register (enable) Portal Server Secure Remote Access components:

   ```
   PortalServer7-base/bin/psadmin provision-sra -u amadminUser
       -f passwordFile -p Portal_ID --gateway-profile profileName --enable
   ```

Upgrading Release 4 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 Release 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 391), except that Release 5 Portal Server Secure Remote Access is installed in the same path as Release 4 on Linux OS. As a result, the `psupgrade` script removes the previous RPMs when installing the Release 5 RPMs.

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 4 to Release 5 on Linux cannot be rolled back. Make sure you back up your system *before* performing the following procedure.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, synchronize all shared components to Release 5.

   Instructions are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63.

   This step is a necessary prerequisite to running the `psupgrade` script in Step 8 on page 395.

3. Stop any instances of the Gateway, Rewriter Proxy, or Netlet Proxy that are running locally.

   ```
   PortalServer6-base/bin gateway stop
   PortalServer6-base/bin netletd stop
   PortalServer6-base/bin rwproxyd stop
   ```

Check that the processes have stopped:

Gateway: `netstat -an | grep 443`
Rewriter Proxy: `netstat -an | grep 10443`
Netlet Proxy: `netstat -an | grep 10555`

4. Make sure Access Manager is running.

5. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script. For example,

   ```
   export ANT_HOME=/opt/sun
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

6. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

7. If the Portal Server Secure Remote Access component you are upgrading is remote from Portal Server, copy the `dpadmin` executable from the computer hosting Portal Server to the computer hosting the Portal Server Secure Remote Access component.

   The dpadmin executable can be found in the following location:

   PortalServer7-base`/SUNWps.bak/bin/dpadmin`, if Portal Server has been upgraded.

   PortalServer6-base`/bin/dpadmin`, if Portal Server has not yet been upgraded.

8. Run the `psupgrade` script from the Java ES Release 5 distribution.

   ```
   cd os_arch/Products/portal_svr/Tools/upgrade/bin
   ./psupgrade
   ```

   where *os_arch* matches your platform, such as `Solaris_sparc`.

   The `psupgrade` script invokes the Java ES installer to install new packages and requests the following information:

   ❍ Directory Server Admin ID and password

   ❍ Access Manager Admin ID and password

   ❍ Directory Server amldapuser ID and password

Verifying the Upgrade

If the Portal Server Secure Remote Access component you are upgrading is remote from Portal Server, you can verify the installation of Release 5 packages by checking the version information in the following file:

```
PortalServer7-base/lib/PSversion.properties
```

However, if the Portal Server Secure Remote Access component you are upgrading is resides on the same computer as Portal Server, you can verify the upgrade using the following command:

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile.
```

See Table 16-4 on page 388 for output values.

You can also check the upgrade log files at:

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log
```

Release 4 Post-Upgrade Tasks

There are no post-upgrade tasks required when upgrading Portal Server Secure Remote Access to Release 5, except for the following situations:

- Restore Configuration for Load Balancer
- Restore Configuration for Directory Proxy Server
- Delete Release 4 Localized Providers

Restore Configuration for Load Balancer

If Portal Server Secure Remote Access instances have been accessed through a load balancer, the following steps need to be performed after upgrade to restore the load balancer configuration:

1. Set the following parameters in the *PortalServer7Config-base*/`platform.conf.default` file:

   ```
   gateway.virtualhost=loadBalancer_hostName loadBalancer_hostIP
   gateway.external.ip=loadBalancer_hostIP
   gateway.dsame.agent=http\://loadBalancer_hostName\:
       80/portal/RemoteConfigServlet
   ```

2. Set the following parameter in the *PortalServer7Config-base*/`GWConfig-default.properties` file.

   ```
   gateway.ipaddress=Gateway_hostIP
   ```

3. Set the parameters corresponding to Step 1 and Step 2 for Rewriter Proxy and Netlet Proxy, when these instances are deployed on computers remote from the Portal Server host.

4. Restart Portal Server and the load-balanced Portal Server Secure Remote Access instances.

Restore Configuration for Directory Proxy Server

If Portal Server Secure Remote Access instances have accessed Directory Server through a Directory Proxy Server instance, the Directory Proxy Server host and port number settings must be restored to their original values before upgrade. See "Remove Configuration for Directory Proxy Server" on page 391, in which the values of these properties were modified in preparation for upgrade.

Delete Release 4 Localized Providers

Localized Proxylet services will not load until you delete the Release 4 localized providers, as follows:

1. Go to the *PortalServer7Data-base*`/portals/Upgraded/desktop` directory.

2. Delete all directories and files from `default_`*Locale* except for:

   - Files and directories you have created (not shipped with Portal Server Secure Remote Access)

   - The `message.properties` file

   - The following directories:

     ```
     AddressBookProvider
     BookmarkProvider
     CalendarProvider
     LoginProvider
     LotusNotesAddressBookProvider
     LotusNotesCalendarProvider
     LotusNotesMailProvider
     MSExchangeAddressBookProvider
     MSExchangeCalendarProvider
     MSExchangeMailProvidervMailProvider
     NotesProvider
     PersonalNoteProvider
     Register
     SampleRSS
     SampleURLScraper
     SampleXML
     TemplateEditContainerProvider
     ```

```
TemplateTabContainerProvider
URLScraperProvider
UWCAddressBookProvider
UserInfo
UserInfoProvider
XMLProvider
error
```

3. Restart the web container. in which Portal Server is deployed.

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 5 consists of reverting back to the Release 4 installation at *PortalServer6-base*.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Restore Directory Server to the state it was in before upgrade.

   Use the Directory Server backup/restore command line and GUI utilities. See the Directory Server Backup and Restore chapter of the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*, http://docs.sun.com/doc/819-0995.

3. Remove the Release 5 Portal Server Secure Remote Access packages.

   a. Launch the Java ES uninstaller.

      ```
      /var/sadm/prod/SUNWentsys5/uninstall
      ```

   b. Select all installed Portal Server Secure Remote Access components.

   c. Confirm your uninstall choice.

   d. Exit the Java ES uninstaller.

4. Restore the *PortalServer6-base* and *PortalServer6Config-base* directories to their original locations.

   During upgrade they were move to directories with a `.bak` extension.

Rolling Back the Upgrade (Linux)

Because the upgrade to Release 5 requires the removal of the Release 4 binaries, it is very difficult to roll back the upgrade on Linux.

One approach to rollback would be to create a parallel system *before* upgrading and testing that system before attempting an upgrade. If you need to roll back the upgrade, you can revert back to that parallel system.

Multiple Instance Upgrades

In some deployment architectures Portal Server Secure Remote Access components, such as Gateway, are deployed on multiple computer systems to provide for security and scalability and to improve availability. For example, you might have Gateway components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Gateway, you can perform a rolling upgrade in which you upgrade Gateway instances sequentially without interrupting service, as described below. The procedure takes into account the following constraint: individual Portal Server Secure Remote Access components are not backwardly compatible with earlier versions; all need to be synchronized, along with Portal Server itself, at Java ES Release 5. However during a rolling upgrade Release 4 Portal Server Secure Remote Access instances can remain running while Portal Server instances are being upgraded.

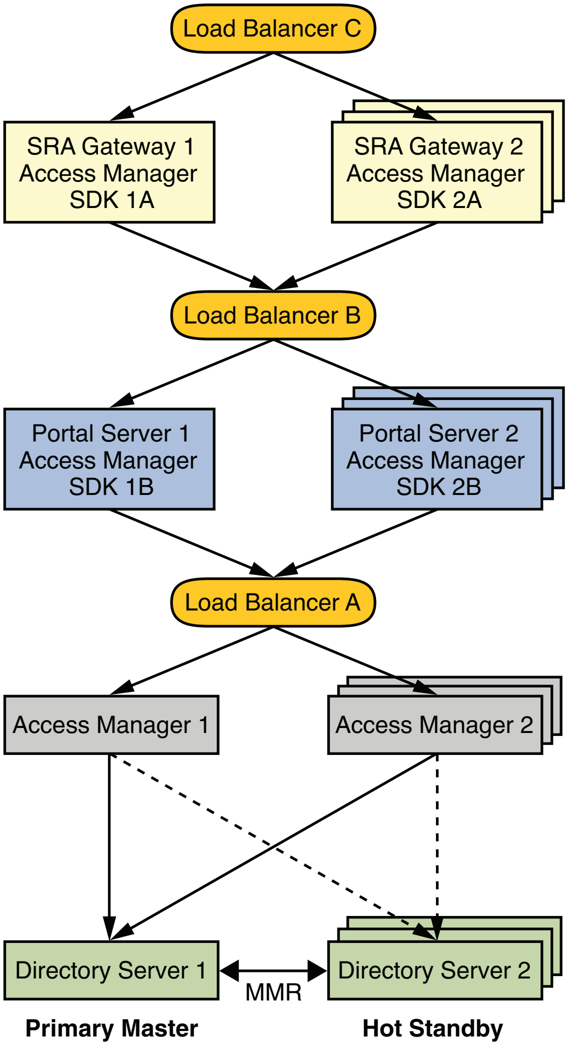
The deployment architecture shown in Figure 16-1 on page 400 will be used to illustrate the rolling upgrade procedure.

In this architecture, multiple Portal Server instances are accessed by way of Portal Server Secure Remote Access Gateway instances. Both the Portal Server instances and the Gateway instances are load balanced to provide for availability and scalability.

The Portal Server instances, in turn, access Access Manager instances through a load balancer. The Access Manager and Access Manager SDK instances access a directory that is set up for multi-master replication (MMR). While other Directory Server replication schemes are possible, MMR is representative of highly available and scalable directory services.

In Figure 16-1, the multiple instances of Gateway, Portal Server, Access Manager, and Directory Server are grouped to facilitate explanation of the upgrade procedure. `Portal Server 2`, for example, is representative of the second through nth instances of Portal Server.

Figure 16-1 Example Deployment Architecture for Multiple Portal Server Instances



Rolling upgrade of Release 4 Gateway (and Portal Server) to Release 5 is performed as follows:

1. If you are upgrading Release 4 Access Manager to Release 5, perform a rolling upgrade as documented in "Multiple Instance Upgrades" on page 285. Note that in upgrading Release 4 Gateway or Release 4 Portal Server to Release 5, you are not required to upgrade Release 4 Access Manager to Release 5.

2. Modify the configuration of Portal Server and Gateway instances as follows.

   a. Configure `Portal Server 2` to point to `Directory Server 2` rather than `Directory Server 1`.

      For brevity, in this and succeeding steps, "`Portal Server 2`" will mean `Portal Server 2` through `Portal Server n`.

   b. Configure `Gateway 2` to point to `Directory Server 2` rather than `Directory Server 1`.

      For brevity, in this and succeeding steps, "`Gateway 2`" will mean `Gateway 2` through `Gateway n`.

3. Upgrade `Portal Server 1`.

   a. Disable `Portal Server 1` in `Load Balancer B`.

      Requests will no longer be routed to `Portal Server 1`.

   b. Disable Directory Server MMR.

      `Directory Server 2` will no longer by synchronized with `Directory Server 1`.

   c. Upgrade `Access Manager SDK 1B` to Release 5.

      Use the procedure in "Release 4 Access Manager SDK-only Upgrades" on page 287.

   d. Upgrade `Portal Server 1` to Release 5.

      Perform the upgrade of the Portal Server instance as described in "Release 4 Portal Server Secure Remote Access Upgrade" on page 387, noting the following:

      - Make special note of the following pre-upgrade task: "Remove Configuration for Load Balancer" on page 390.
      - Confirm, before performing the upgrade, that the value of `am.encryption.pwd` in the *AccessManagerConfig-base*`/config/AMConfig.properties` file is the same for the local Access Manager SDK as for its associated remote Access Manager instance.
      - Make sure that you provide a non-null, unique value for the `Portal Instance ID` parameter requested by `psupgrade` for each Portal Server instance that you are upgrading.

      Portal Server data for `Directory Server 1` is updated to Release 5.

4. Upgrade `Gateway` 1.

   a. Disable `Gateway` 1 in `Load Balancer` C.

      Requests will no longer be routed to `Gateway` 1.

   b. Upgrade `Access Manager SDK 1A` to Release 5.

      Use the procedure in "Release 4 Access Manager SDK-only Upgrades" on page 287.

   c. Upgrade `Gateway` 1 to Release 5.

      Perform the upgrade of Gateway as described in "Release 4 Portal Server Secure Remote Access Upgrade" on page 387, noting the following:

      - Make special note of the following pre-upgrade task: "Remove Configuration for Load Balancer" on page 390.

      - Confirm, before performing the upgrade, that the value of `am.encryption.pwd` in the *AccessManagerConfig-base*`/config/AMConfig.properties` file is the same for the local Access Manager SDK as for its associated remote Access Manager instance.

5. Enable the previously disabled `Portal Server` 1 and `Gateway` 1 in their respective load balancers, as follows:

   a. Enable `Portal Server` 1 in `Load Balancer` B.

      Requests will be once again routed to `Portal Server` 1.

   b. Enable `Gateway` 1 in `Load Balancer` C.

      Requests will be once again routed to `Gateway` 1.

6. Disable `Portal Server` 2 and `Gateway` 2 in their respective load balancers, as follows:

   a. Disable `Portal Server` 2 in `Load Balancer` B.

      Requests will no longer be routed to `Portal Server` 2.

   b. Disable `Gateway` 2 in `Load Balancer` C.

      Requests will no longer be routed to `Gateway` 2.

7. Upgrade `Portal Server 2`.

   a. Restore the configuration of `Portal Server 2` to point to `Directory Server 1`.

   b. Upgrade `Access Manager SDK 2B` to Release 5.

      Use the procedure in "Release 4 Access Manager SDK-only Upgrades" on page 287.

   c. Upgrade `Portal Server 2` to Release 5.

      Use the same procedure as in Upgrade `Portal Server 1`, Step d on page 401.

   d. Enable `Portal Server 2` in `Load Balancer B`.

      Requests will be once again routed to `Portal Server 2`.

8. Upgrade `Gateway 2`.

   a. Restore the configuration of `Gateway 2` to point to `Directory Server 1`.

   b. Upgrade `Access Manager SDK 2A` to Release 5.

      Use the procedure in "Release 4 Access Manager SDK-only Upgrades" on page 287.

   c. Upgrade `Gateway 2` to Release 5.

      Use the same procedure as in Upgrade `Gateway 1`, Step c on page 402.

   d. Enable `Gateway 2` in `Load Balancer C`.

      Requests will be once again routed to `Gateway 2`.

9. Enable Directory Server MMR.

   The Portal Server data for `Directory Server 2`, is now synchronized with `Directory Server 1`.

NOTE In rolling upgrades scenarios in which Portal Server instances are being upgraded to Release 5 while earlier releases of the Gateway component remain active (which is *not* the case in the above procedure), and in which Gateway instances are accessed through a load balancer, you should check for all Gateway instances that the following configuration properties in the *PortalServer6Config-base*/`GWConfig.properties` file and `GWConfig-default.properties` file reference the Gateway and *not* the load balancer:

`GW_IP=`*Gateway_hostIP*
`GW_HOST=`*Gateway_hostName*

If these properties point to the load balancer, the Gateway will no longer access upgraded Portal Server instances.

Upgrading Portal Server Secure Remote Access from Java ES Release 3

The procedure for upgrading Java ES 2005Q1 (Release 3) Portal Server Secure Remote Access to Release 5 is the same as that for upgrading Release 4 Portal Server Secure Remote Access to Release 5, with the following exceptions:

- Upgrading Portal Server Secure Remote Access Dependencies
- Upgrading Release 3 Portal Server Secure Remote Access
- Multiple Instance Upgrades

Upgrading Portal Server Secure Remote Access Dependencies

However, when upgrading Portal Server Secure Remote Access from Release 3, you have to upgrade Access Manager to Release 4 or to Release 5 before upgrading Portal Server Secure Remote Access, and you cannot leave any other dependencies at Release 3, nor upgrade some dependencies to Release 4 and others to Release 5. For more information, see “Selective Upgrade Issues” on page 385.

The following dependencies need to be upgraded in the order shown below.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, “Upgrading Java ES Shared Components” on page 63.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 5 are provided in “Upgrading Directory Server from Java ES Release 2” on page 115.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 5 are provided in Chapter 14, “Access Manager” on page 261.
4. **Portal Server.** Instructions for upgrading Portal Server are provided in Chapter 15, “Portal Server” on page 309.

Upgrading Release 3 Portal Server Secure Remote Access

To upgrade Release 3 Portal Server Secure Remote Access to Release 5, use the instructions in "Upgrading Portal Server Secure Remote Access from Java ES Release 4" on page 386, except substitute Release 3 wherever Release 4 is referenced.

Multiple Instance Upgrades

In some deployment architectures Portal Server Secure Remote Access components, such as Gateway, are deployed on multiple computer systems to provide for security and scalability and to improve availability. For example, you might have Gateway components running on multiple computers with a load balancer to distribute the load.

When performing multiple instance upgrades from Release 3 Portal Server Secure Remote Access, use the procedure documented in "Multiple Instance Upgrades" on page 399, except replace "Release 4" with "Release 3" wherever Release 4 is referenced. You must also upgrade Access Manager, as described in Step 1 on page 400.

Upgrading Portal Server Secure Remote Access from Java ES Release 2

This section includes information about upgrading Java ES 2004Q2 (Release 2) Portal Server Secure Remote Access to Release 5. The upgrade procedure is similar to that for upgrading Release 4 Portal Server Secure Remote Access to Release 5, except for some changes as documented in the following sections:

- Release 2 Pre-Upgrade Tasks
- Upgrading Release 2 Portal Server Secure Remote Access
- Release 2 Post-Upgrade Tasks
- Multiple Instance Upgrades

NOTE If you are upgrading from Release 2 Portal Server Secure Remote Access on the Linux platform, then you will have to perform a dual upgrade, in which both Portal Server Secure Remote Access *and* the operating system are upgraded (Release 5 Portal Server Secure Remote Access is not supported on RHEL 2.1). See "Dual Upgrade" on page 385 for more information.

Release 2 Pre-Upgrade Tasks

The pre-upgrade tasks for upgrading Portal Server Secure Remote Access from Release 2 are the same as those documented in "Release 4 Pre-Upgrade Tasks" on page 388, except for the following tasks:

- Upgrading Portal Server Secure Remote Access Dependencies
- Delete Gateway Service Entry

Upgrading Portal Server Secure Remote Access Dependencies

When upgrading Portal Server Secure Remote Access from Release 2, you have to upgrade Access Manager to Release 4 or to Release 5 before upgrading Portal Server Secure Remote Access, and you cannot leave any other dependencies at Release 2, nor upgrade some dependencies to Release 4 and others to Release 5. For more information, see "Selective Upgrade Issues" on page 385.

The following dependencies need to be upgraded in the order shown below.

1. **Shared Components.** Instructions for upgrading Java ES shared components to Release 5 are provided in Chapter 2, "Upgrading Java ES Shared Components" on page 63. However, if shared components have not yet been upgraded, they will be upgraded automatically by the psupgrade script.
2. **Directory Server.** Instructions for upgrading Directory Server to Release 5 are provided in "Upgrading Directory Server from Java ES Release 2" on page 115.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to Release 5 are provided in Chapter 14, "Access Manager" on page 261.
4. **Portal Server.** Instructions for upgrading Portal Server are provided in Chapter 15, "Portal Server" on page 309.

To upgrade Release 2 Portal Server Secure Remote Access to Release 5, use the instructions in "Upgrading Portal Server Secure Remote Access from Java ES Release 4" on page 386, except substitute Release 2 wherever Release 4 is referenced.

Delete Gateway Service Entry

The amService-srapGateway user entry must be manually deleted when upgrading Portal Server from Release 2, otherwise the Gateway will fail to start after upgrade. Perform the following steps:

1. Log in to Access Manager Console.
2. List all users in the organization DN.
3. Delete the amService-srapGateway user.

Upgrading Release 2 Portal Server Secure Remote Access

The procedure for upgrading Java ES 2004Q2 (Release 2) Portal Server Secure Remote Access to Release 5 is the same as for upgrading Release 4 Portal Server Secure Remote Access to Release 5.

To upgrade Release 2 Portal Server Secure Remote Access to Release 5, use the instructions in "Upgrading Portal Server Secure Remote Access from Java ES Release 4" on page 386, except substitute Release 2 wherever Release 4 is referenced.

Release 2 Post-Upgrade Tasks

The post-upgrade tasks for upgrading from Release 2 are the same as those documented in "Release 4 Post-Upgrade Tasks" on page 396, except for the following task:

- Set Portal Server Domain for Proxylet Service

Set Portal Server Domain for Proxylet Service

After upgrading Release 2 Portal Server Secure Remote Access to Release 5, you have to set the correct Portal Server domain value.

1. Log in to Portal Server Console, and navigate to the Proxylet tab under Secure Remote Access.
2. Select the distinguished name (DN) of the Organization where the Proxylet service is found.
3. Under the Domains field of Proxylet rules, replace `SERVER_DOMAIN` with the domain name where Portal Server is installed.
4. Repeat the above steps for all organizations where Proxylet is service is found.

Multiple Instance Upgrades

Multiple instance rolling upgrades (see"Multiple Instance Upgrades" on page 399) are not supported in upgrading Release 2 Portal Server Secure Remote Access components (or Portal Server) to Release 5.

Upgrading Portal Server Secure Remote Access from the Interim Feature Release 7.0

This section includes information about upgrading Portal Server Secure Remote Access from the Interim Feature Release (IFR) 7.0 2005Q4 to Java ES 5 (Release 5).

The section covers the following topics:

- Introduction
- Portal Server Secure Remote Access IFR 7.0 Upgrade
- Multiple Instance Upgrades

Introduction

When upgrading Portal Server Secure Remote Access IFR 7.0 to Release 5, consider the following aspects of the upgrade process:

The psupgrade script for upgrading Portal Server Secure Remote Access IFR to Release 5 does not install new packages, as in the case of upgrade from Release 4. Instead, the upgrade procedure will require you to apply the following patches:

Table 16-5 Patches[1] to Upgrade Portal Server Secure Remote Access IFR to Release 5

Description	Patch ID: Solaris 9 & 10	Patch ID: Linux
Portal Server 7.1	121465-10 (SPARC) 121466-10 (x86)	121467-10
Portal Server 7.1 localization	123254-02 (SPARC) 124590-02 (x86)	123255-02

[1] Patch revision numbers are the minimum required for upgrade to Java ES Release 5. If newer revisions become available, use the newer ones instead of those shown in the table.

Portal Server Secure Remote Access IFR 7.0 Upgrade

This section describes how to perform an upgrade of Portal Server Secure Remote Access from the IFR to Java ES Release 5 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
- Upgrading Portal Server Secure Remote Access IFR 7.0 (Solaris)
- Upgrading Portal Server Secure Remote Access IFR 7.0 (Linux)
- Verifying the Upgrade
- Post-Upgrade Tasks
- Rolling Back the Upgrade (Solaris)
- Rolling Back the Upgrade (Linux)

Pre-Upgrade Tasks

Pre-upgrade tasks for the IFR upgrade are the same as for the Release 4 upgrade (see "Release 4 Pre-Upgrade Tasks" on page 388).

Upgrading Portal Server Secure Remote Access IFR 7.0 (Solaris)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

IFR 7 Upgrade Considerations (Solaris)

The Portal Server Secure Remote Access IFR upgrade to Release 5 takes into account the same considerations as the Release 4 upgrade (see "Upgrade Considerations (Solaris)" on page 391).

IFR 7 Upgrade Procedure (Solaris)

The procedure documented below applies to Portal Server Secure Remote Access on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

```
su -
```

2. Stop any instances of the Gateway, Rewriter Proxy, or Netlet Proxy that are running locally.

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t gateway  -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t rwproxy -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t nlproxy -N gatewayProfileName
   ```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

3. Make sure Access Manager is running.

4. Obtain the required patch, based on Table 16-5 on page 410.

   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`

5. Apply the appropriate Portal Server patch and, if needed, localization patch in Table 16-5.

   ```
   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. In cases where localization packages have been upgraded in Step 5, set the Portal Server Console JVM's locale to UTF-8.

   ```
   export LC_ALL=ja_JP.UTF-8
   export LANG=ja_JP.UTF-8
   ```

8. Set two environment variables (`ANT_HOME` and `JAVA_HOME`) needed by the `psupgrade` script:

   ```
   export ANT_HOME=/usr/sfw
   export JAVA_HOME=/usr/jdk/entsys-j2se
   ```

9. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

10. Run the psupgrade script.

    ```
    cd PortalServer7-base/bin
    ./psupgrade
    ```

    The psupgrade script is not run from the Java ES Release 5 distribution and does not invoke the Java ES installer (the packages were already patched).

Upgrading Portal Server Secure Remote Access IFR 7.0 (Linux)

This section discusses considerations that impact the upgrade procedure for Portal Server Secure Remote Access followed by a description of the procedure itself.

IFR 7 Upgrade Considerations (Linux)

The upgrade of Portal Server Secure Remote Access software to Release 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 391), except that installing the Release 5 patches on Linux OS removes the previous RPMs.

IFR 7 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 Portal Server Secure Remote Access IFR to Release 5 on Linux cannot be rolled back. Make sure you back up your system *before* performing the following procedure.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop any instances of the Gateway, Rewriter Proxy, or Netlet Proxy that are running locally.

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t gateway  -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t rwproxy -N gatewayProfileName
   ```

   ```
   PortalServer7-base/bin/psadmin stop-sra-instance -u amadminUser
      -f passwordFile -t nlproxy -N gatewayProfileName
   ```

   Check that the processes have stopped:

   Gateway: `netstat -an | grep 443`
   Rewriter Proxy: `netstat -an | grep 10443`
   Netlet Proxy: `netstat -an | grep 10555`

3. Make sure Access Manager is running.

4. Obtain the required patch using the patch numbers and RPM names from Table 16-5 on page 410.

   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`

5. Apply the Portal Server patch and, if needed, localization RPMs for Portal Server in Table 16-5, in that order.

   See the Readme file for the Portal Server patch, which describes how to use a script to apply the patch's RPMs:

   ```
   cd /tmp
   ```

   where /tmp is the directory to which you download the patch.

   ```
   ./update
   ```

   The update script installs the RPM's.

   For the localization patch, install each RPM using the following command:

   ```
   rpm -Fvh patchName-version.rpm
   ```

6. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-portal-core
   ```

   The upgrade revision numbers of the RPMs should be returned.

7. In cases where localization packages have been upgraded in Step 5, set the Portal Server Console JVM's locale to UTF-8.

```
export LC_ALL=ja_JP.UTF-8
export LANG=ja_JP.UTF-8
```

8. Set two environment variables (ANT_HOME and JAVA_HOME) needed by the psupgrade script:

```
export ANT_HOME=/opt/sun
export JAVA_HOME=/usr/jdk/entsys-j2se
```

9. Make sure you have adequate swap space on your computer.

   As a guideline, the swap space should be set to twice the amount of physical ram.

10. Run the psupgrade script.

```
cd PortalServer7-base/bin
./psupgrade
```

   The psupgrade script is not run from the Java ES Release 5 distribution and does not invoke the Java ES installer (the packages were already patched).

Verifying the Upgrade

You can verify the patching of Portal Server Secure Remote Access packages to Release 5 using the following command:

```
PortalServer7-base/bin/psadmin --version --adminuser admin_ID
-f adminpasswordfile.
```

See Table 16-4 on page 388 for output values.

You can also check the upgrade log files at:

```
/var/sadm/install/logs/Sun_Java_System_Portal_Server_upagrede.log
```

Post-Upgrade Tasks

There are no post-upgrade tasks required when upgrading Portal Server Secure Remote Access to Release 5.

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 5 consists of reverting back to the IFR installation at *PortalServer7-base*.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Restore Directory Server to the state it was in before upgrade.

   Use the Directory Server backup/restore command line and GUI utilities. See the Directory Server Backup and Restore chapter of the *Sun Java System Directory Server Enterprise Edition 6.0 Administration Guide*, `http://docs.sun.com/doc/819-0995.`

3. Back out the Portal Server 7.1 patch in Table 16-5 on page 410.

   ```
   patchrm patch_ID
   ```

Rolling Back the Upgrade (Linux)

On the Linux platform there is no procedure for rolling back the upgrade.

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 "Portal Server Secure Remote Access IFR 7.0 Upgrade" on page 411.

When performing multiple instance upgrades from IFR Portal Server Secure Remote Access, use the procedure documented in "Multiple Instance Upgrades" on page 399, except replace "Release 4" with "IFR" wherever Release 4 is referenced. You must also upgrade Access Manager, as described in Step 1 on page 400.

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 2004Q2 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 OS.

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 2004Q2 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 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 and Sun Cluster Agents are supported on the Solaris OS but not on the Linux OS.

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 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)

Java ES 5 (Release 5)

This section lists the contents of Java Enterprise System 5.

Release 5 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. Note that Communication Services components are no longer supported by the Java ES installer.

Collaboration Services

- Sun Java System Portal Server 7.1
- Sun Java System Portal Server Secure Remote Access 7.1
    - Gateway
    - Netlet Proxy
    - Rewriter Proxy

Web & Application Services

- Sun Java System Application Server Enterprise Edition 8.2
    - 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.
    - Sample Applications
- Sun Java System Web Server 7.0
    - Web Server 7.0 CLI
    - Web Server 7.0 Core
    - Web Server 7.0 Samples
- Sun Java Web Proxy Server 4.0.4

- Sun Java System Message Queue 3.7UR1
- Service Registry 3.1
    - Service Registry Client Support
    - Service Registry Deployment Support

Directory & Identity Services

- Sun Java System Access Manager 7.1
    - Identity Management and Policy Services Core
    - Access Manager Administration Console
    - Common Domain Services for Federation Management
    - Access Manager SDK
    - Distributed Authentication
    - Client SDK
    - Session Failover Client
- Sun Java System Directory Server Enterprise Edition 6
    - Java Enterprise System Directory Server 6 Core Server
    - Java Enterprise System Directory Service Control Center
    - Sun Java System Directory Server Enterprise Edition 6 Command-Line Utilities
    - Sun Java System Directory Proxy Server 6 Core Server

Availability Services

- Sun Cluster 3.1
    - Sun Cluster Core
- Sun Cluster Geographic Edition 3.1 2006Q4
- Sun Cluster Agents 3.1
    - HA Sun Java System Application Server
    - HA Sun Java System Message Queue
    - HA Sun Java System Directory Server
    - HA Sun Java System Messaging Server

  - HA Sun Java System Application Server EE (HADB)
  - HA/Scalable Sun Java System Web Server
  - HA Instant Messaging
  - HA Sun Java System Calendar Server
  - ...
- Sun Java System High Availability Session Store 4.4.3

Shared Services

- All Shared Components

  See “Release 5 Shared Components” on page 436
- Sun Java System Monitoring Console 1.0
- Java DB
  - Java DB Client
  - Java DB Server

NOTE Sun Cluster and Sun Cluster Agents are supported on the Solaris OS but not on the Linux OS.

Release 5 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) for Sun Cluster only
- CAC (Common agent container)
- FIS (FastInfoSet)
- 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 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
- JAXWS (Java API for Web Services) Runtime
- JDMK (Java Dynamic Management™ Kit) Runtime
- JSS (Java Security Services)
- JSTL (JSP Standard Library Template)
- 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)
- XWSS (XML Web Services Security)

Index