Sun Java™ System

Sun Java Communications Suite 5 Upgrade Guide

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

Part No: 819-7561-10

Contents

List of Tables

Preface

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

This book assumes you are familiar with the following:

- Installation of enterprise-level software products
- Communications Suite components currently deployed in your environment
- Java Enterprise System components currently deployed in your environment
- System administration and networking on your supported Communications Suite platform
- Clustering model (if you are installing clustering software)
- Solaris Zones (if you are deploying into a zoned environment)

Conventions Used in This Book

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

Typographic Conventions

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

Table 1 Typographic Conventions

Typeface	Meaning	Examples
AaBbCc123 (Monospace)	API and language elements, HTML tags, web site URLs, command names, file names, directory path names, onscreen computer output, sample code.	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
AaBbCc123 (Monospace bold)	What you type, when contrasted with onscreen computer output.	`% su` `Password:`

Table 1 Typographic Conventions *(Continued)*

Typeface	Meaning	Examples
AaBbCc123 (Italic)	Book titles, new terms, words to be emphasized. A placeholder in a command or path name to be replaced with a real name or value.	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. The file is located in the *install-dir*`/bin` directory.

Symbols

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

Table 2 Symbol Conventions

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

Shell Prompts

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

Table 3 Shell Prompts

Shell	Prompt
C shell on UNIX or Linux	*machine-name*%
C shell superuser on UNIX or Linux	*machine-name*#
Bourne shell and Korn shell on UNIX or Linux	$
Bourne shell and Korn shell superuser on UNIX or Linux	#

Related Documentation

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

You should review the *Java Enterprise System Upgrade Guide* particularly if you are upgrading non-Communications Suite Java Enterprise System products that are not covered in this guide.

Books in This Documentation Set

The books in this documentation set 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 Communications Suite documentation includes information about the system as a whole and information about its components. This documentation can be accessed here:

http://docs.sun.com/app/docs/prod/sunjava.comm

Table 4 on page 15 lists the manuals in the Communications Suite 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 Communications Suite Documentation

Document	Contents
Sun Java Communications Suite 5 Release Notes http://docs.sun.com/doc/819-4432	Contains the latest information about all Communications Suite component products, including known problems.
Sun Java Communications Suite 5 Installation Guide http://docs.sun.com/doc/819-7560	Guides you through the process of installing Communications Suite and those Java Enterprise System components on which Communications Suite products rely, on the Solaris™ Operating System or the Linux operating system. Also shows how to configure components after installation, and verify that they function properly.
Sun Java Communications Suite 5 Upgrade Guide http://docs.sun.com/doc/819-7561	Provides instructions for upgrading Communications Suite on the Solaris™ Operating System and the Linux operating environment.
Sun Java Communications Suite 5 Deployment Planning Guide http://docs.sun.com/doc/819-4439	Provides deployment information for Communications Suite products.

Accessing Sun Java 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 Communications Suite and its components:

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

Contacting Sun Java Technical Support

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

Related Third-Party Web Site References

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to http://docs.sun.com and click Send Comments. In the online form, provide the document title and part number. The part number for this edition of the *Sun Java Communications Suite Upgrade Guide* is 819-7561-10.

Chapter 1

Planning for Upgrades

This chapter provides information used for planning the upgrade from previous versions of Sun Java™ Communications Suite product components to Communications Suite 5 in the following sections:

Communications Suite 5 Components

As an introduction to planning the upgrade of Communications Suite software, this section reviews the components included in this release of Communications Suite. Depending on your upgrade scenario, you might need to upgrade one or more of these components.

Communications Suite components interoperate with Java Enterprise System components. These component sets are grouped into different types. Accordingly, service components provide the main infrastructure services, while service quality components enhance those system services. These two types of components are together referred to here as *product* components. Each product component depends on one or more locally shared libraries known as *shared* components.

For instructions on upgrading Java Enterprise System components see the *Java Enterprise System Upgrade Guide* (`http://docs.sun.com/doc/819-6553`).

Product Components

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

Table 1-1 Communications Suite Product Components

Product Component	Version	Type	Short Name
Calendar Server	6.3	System service component	CS
Communications Express	6.3	Service quality: access component	CX
Connector for Microsoft Outlook	7.2	Service quality: client component	OC
Delegated Administrator	6.4	Service quality: administrative component	DA
Instant Messaging	7.2	System service component	IM
Messaging Server	6.3	System service component	MS

Instructions for upgrading Connector for Microsoft Outlook are available in the Sun Java System *Connector for Microsoft Outlook Installation Guide*.

Communications Suite Shared Components

The Communications Suite shared components are listed in the following table.

Table 1-2 Communications Suite Shared Components

Shared Component	Version	Abbreviation
Common agent container	2.0	CAC (CACAO)
International Components for Unicode	3.2	ICU
Instant Messaging SDK	7.2	IM SDK
Java 2 Platform, Standard Edition	5.0 Update 6	J2SE™
Java API for XML Processing	1.3.1	JAXP
Java Development Kit	1.5.0_09	JDK
Java Security Services (Network Security Services for Java)	4.2.4 and 3.1.11	JSS and JSS3
JavaBeans Activation Framework	1.0.3	JAF
Java Calendar API	1.2	JCAPI
Java Dynamic Management Kit	5.1	JDMK
Java Mail	1.3.2	Java Mail
LDAP Java SDK	4.19	LDAP J SDK (LDAP JDK)
C++ Libraries	-	LIBCPLUSPLUS
Multi-threaded Memory Allocator Library	-	;LIBMTMALLOC
Sun Java Monitoring Framework	2.0	MFWK
Netscape Portable Runtime	4.6.4	NSPR
Network Security Services	3.11.4	NSS
Small Collection of Programs that Operate on Patch Files	-	PATCHUTILS
Install Software from a Server to a Target Host	-	PKGINSTALL
Simple Authentication and Security Layer	2.19	SASL

About Communications Suite Upgrades

There is no single system utility that upgrades all Communications Suite components. Instead, the upgrade of Communications Suite product components is performed component-by-component, computer-by-computer, using component-specific upgrade procedures documented in this *Upgrade Guide*.

The upgrade of a component can range from a major upgrade, which might not be compatible with the previous version of the component, to a fully-compatible upgrade that simply provides bug fixes. Because of dependencies between Communications Suite components and between Communications Suite and Java Enterprise System components, the nature of the upgrade can impact whether you need to upgrade other components as well.

Product Component Upgrades

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

- **Installing upgraded software.** The new software can enhance or fix existing software, or replace existing software. In general, the new software is achieved through the application of patches to existing software packages, the selective replacement of existing packages or the installation of new packages.
- **Re-configuration.** Re-configuration encompasses any change in configuration data, user data, or dynamic application data needed to support the upgraded software. A change in data can mean additional data, a change in data format (whether in property files or database schema), or a change in data location. Sometimes re-configuration requires that you perform an explicit procedure and sometimes it takes place automatically without your involvement. In some cases, re-configuration also requires re-deployment of component software to a web container.

Both of these aspects of component upgrade are described in this *Upgrade Guide* for each of the Communications Suite product components.

This book also covers other important aspects of product component upgrade, including:

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

Product Component Upgrade Approaches

The upgrade of each product component, as documented in this *Upgrade Guide*, involves one of the following upgrade approaches:

- Running a Component-specific Upgrade Script
- Patching Existing Component Packages

Each of these approaches is discussed briefly in the sections that follow.

Running a Component-specific Upgrade Script

A number of product components provide an upgrade script for automating the upgrade of the component to Communications Suite 5. The script generally performs both the upgrade of software packages and any re-configuration required as part of the upgrade. For those components deployed to a web container, the script generally re-deploys the upgraded component software to the web container.

Patching Existing Component Packages

For a number of product components, upgrade is performed by manually patching existing software packages. Solaris and Linux platforms employ similar technologies for managing installed software packages and tracking changes to those packages through a package registry.

- **Solaris platform.** Packages can be 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 website at: `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

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

- **Linux platform.** RPM (Red Hat Package Manager) packages 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 Communications Suite 5 distribution, but also through the SunSolve website at: `http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access`

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

Shared Component Upgrades

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

The upgrading of shared components is typically more straightforward than the upgrading of product components: there is normally no reconfiguration required, no pre or post upgrade procedures to be performed, and no rollback supported.

Operating System Issues

A number of operating system issues impact the upgrading of Communications Suite software, as described below.

Required Operating System Updates

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

- Solaris platform patches are available through the SunSolve website as a patch cluster, a collection of operating system patches that can be collectively applied. The operating system patch clusters for Solaris 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/

Operating System Upgrades

Upgrading an operating system can have an impact on installed Communications Suite components, both product components and shared components, often requiring that at least some components be restored or upgraded. Upgrading the operating system can affect Communications Suite components in the following two ways:

- **Release-specific packages.** A significant number of Communications Suite shared components have Solaris release-specific packages that do not function correctly on other Solaris platform releases. For example, packages that are released for the Solaris 9 operating system cannot always be expected to work on Solaris 10 operating system.

  When upgrading the operating system from one release to another, the various installed Communications Suite shared components that have release-specific packages will need to be upgraded to match the newly upgraded operating system.

- **Operating system-bundled shared components.** In some cases the updated version of operating system includes versions of Communications Suite shared components earlier than those required by the installed Communications Suite software. In other words, upgrading the operating system overwrites the existing Communications Suite shared components with earlier versions. In that case the correct Communications Suite versions need to be restored.

Upgrading Non-Supported Systems

There are two situations by which an operating system and Communications Suite software could become misaligned:

- You upgrade the operating system to a version not supported by the installed Communications Suite software.

  For example, Java Enterprise System 2004Q2 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 Enterprise System 2004Q2, you will also need to upgrade your Java Enterprise System 2004Q2 to a Communications Suite release version that supports the upgraded platform, preferably to Communications Suite 5.

Because upgrade of some components requires that other Communications Suite components be running, you cannot, as a general rule, simply upgrade your operating system platform to Solaris 10 or RHEL 3.0 (which are not supported by Communiations Services 2004Q2) and then upgrade product components from 2004Q2.

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

  For example, Java Enterprise System 2005Q1 and Java Enterprise System 2005Q4 are supported on Solaris 8 and RHEL 2.1. If you want to upgrade to Communications Suite 5, however, which is not supported on Solaris 8 or RHEL 2.1, you will also need to upgrade your operating system to versions supported by Communications Suite 5, preferably to Solaris 10 or RHEL 4.0.

In both these situations, you have to upgrade both Communications Suite software and operating system software.

In general, such dual upgrades require the following upgrade sequence:

1. Back up all Communications Suite configuration files and customizations.
2. Uninstall the currently-installed Communications Suite release version.
3. Upgrade your operating system software.
4. Perform a fresh install of Communications Suite 5. See the *Communications Suite Installation Guide.*
5. Configure Communications Suite 5 using saved configuration data.

Upgrade Planning

The approach you take to upgrading to Communications Suite 5 can depend on your upgrade objectives and priorities, as well as the scope and complexity of your deployment architecture.

For example, your Communications Suite deployment architecture might consist of a single Communications Suite 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 deployment architecture might consist of a number of interdependent Communications Suite components deployed across a number of different computers, and your upgrade objective is to achieve some new functionality by upgrading the minimum number of components required to achieve that end with minimal downtime.

These two examples represent upgrade scenarios of very different complexity, requiring substantially different upgrade plans. No one plan works for all deployed Communications Suite software systems.

In general, the greater the number of Communications Suite components and the greater the number of computers in your deployment architecture, the more complex will be your upgrade plan. This section discusses considerations that impact an upgrade plan.

What is an Upgrade Plan?

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

Table 1-3 Phases in the Upgrade Process

Upgrade Phase	Description
Preparation	You develop an upgrade plan. In it, you specify the Communications components you need to upgrade and the sequence by which you need to upgrade those components on the various computers in your system. You also plan how to test upgrade procedures in a staging environment before executing them in your production environment. In this step, you also back up your current system and test your ability to restore it to its current configuration. You might also write scripts to be executed by IT personnel.
Execution	You obtain all the necessary packages, patches, and tools needed for the upgrade. You execute the upgrade and re-configuration of your Communications Suite deployed system in a staging environment. This involves the backup of configuration and application data, the upgrade of system software, and the re-configuration or migration of data to the upgraded system.
Verification	You start the upgraded software components and perform verification tests as you proceed. If verification is not successful, and problems cannot be resolved within a reasonable time frame, you might be forced to roll back the upgrade and restore the system to its previous state.
Rollback/restoration	If necessary, you restore the system to its previous state as specified in the preparation phase. You also perform tests to verify that the rollback is successful.

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

Upgrade Plan Considerations

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

- Upgrade path
- Dependencies between deployed Communications Suite components
- Selective upgrade
- Multi-instance upgrades

These factors are discussed in the following sections.

Upgrade Paths

While it is possible to upgrade all previous releases of Communications Suite software to Communications Suite 5, the only certified upgrades are from Java Enterprise System 2005Q4, Java Enterprise System 2005Q1, and Java Enterprise System 2004Q2. Upgrades from earlier releases are not documented in this *Upgrade Guide*.

The various upgrade paths involve different upgrade strategies, as described in Table 1-4 on page 26.

Table 1-4 Upgrade Paths to Communications Suite 5

Product Number	Communications Suite Release	System Characteristics	Upgrade Strategies
2005Q4	Java Enterprise System 2005Q4	Communications Suite 5 supports a mixture of 2005Q4 and 5 components on a single computer. This includes both product components and shared components. Compatibilities between 2005Q4 and Communications Suite 5 components have been tested, and any known incompatibilities are noted in the *Sun Java Communications Suite Release Notes*.	The coexistence of 2005Q4 and Communications Suite 5 components provides for the possibility of selectively upgrading 2005Q4 components to Communications Suite 5 on a given computer, or within a deployment architecture consisting of multiple computers.

Table 1-4 Upgrade Paths to Communications Suite 5 *(Continued)*

Product Number	Communications Suite Release	System Characteristics	Upgrade Strategies
2005Q1	Java Enterprise System Release 3	Communications Suite 5 supports a mixture of 2005Q1 and Communications Suite 5 components on a single computer. This includes both product components and shared components. Compatibilities between 2005Q1 and 5 components have been tested, and any known incompatibilities are noted in the in the *Sun Java Communications Suite Release Notes*.	The coexistence of 2005Q1 and Communications Suite 5 components provides for the possibility of selectively upgrading 2005Q1 components to Communications Suite 5 on a given computer, or within a deployment architecture consisting of multiple computers.
2004Q2	Java Enterprise System Release 2	Communications Suite 5 does not support a mixture of 2004Q2 and Communications Suite 5 components on a single computer. This includes both product components and shared components. There are known incompatibilities between the release versions, and interoperability between 2004Q2 and Communications Suite 5 components is not certified.	When upgrading components from 2004Q2 to Communications Suite 5 on a given computer, all 2004Q2 components should be upgraded to Communications Suite 5. However, assuming compatibility of components, it is possible to mix 2004Q2 and Communications Suite 5 components residing on *different* computers within a deployment architecture consisting of multiple computers.
2003Q4 and prior	Java Enterprise System Release 1 and prior	Communications Suite 5 does not support a mixture of Release 1 or prior releases and 5 components on a single computer. This includes both product components and shared components. There are known incompatibilities between the release versions, and interoperability between Release 1 or prior components and Communications Suite 5 components is not certified.	Communications Suite 5 does not certify the direct upgrade of Release 1 or prior releases to 2005Q4. In some cases, however, It is possible to perform an upgrade from Release 1 by upgrading first to Java Enterprise System 2005Q1, as documented in the 2005Q1 *Java Enterprise System Upgrade and Migration Guide* (`http://docs.sun.com/doc/819-0062`). In other cases the upgrade from Release 1 to 5 can be performed in the same way as the upgrade from 2004Q2 or 2005Q1 to 5, and in those cases, the upgrade roadmap for that component in this *Upgrade Guide* notes this possibility.

NOTE Some product components have issued interim releases that fall between the official Communications Suite releases. In such cases the upgrade of the interim release should be performed using the same procedure as for the previous Communications Suite release. For example, if an interim release took place between 2004Q2 and 2005Q1, the component would be upgraded using the procedure for upgrading from 2004Q2 to Communications Suite 5.

Upgrade Dependencies

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

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

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

Upgrading a Communications Suite component requires you to upgrade all the components upon which it has hard upgrade dependencies, but allows you to not upgrade components upon which it has soft upgrade dependencies. (This general rule does not apply to upgrades from 2004Q2 to Communications Suite 5 on a single computer.)

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

Selective Upgrade or Upgrade All

The difference between hard and soft upgrade dependencies allows for the possibility of selectively upgrading Communications Suite components within a deployed system. This possibility only applies to upgrading from 2005Q1 and 2005Q4 to Communications Suite 5 on a single computer (see upgrade path characteristics in "Upgrade Paths" on page 26). Selective upgrade from 2004Q2 to Communications Suite 5 on a single computer is not supported.

- **Selective Upgrade.** The selective upgrade approach starts with the Communications Suite component you wish to upgrade to Communications Suite 5. Determine the hard upgrade dependencies for that component, which includes dependencies on both product components and shared components. Those components also need to be upgraded. You repeat this process for each successive hard upgrade dependency until no further components need to be upgraded. This exercise specifies all Communications Suite components that need to be upgraded.

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

- **Upgrade All.** As an alternative, you can upgrade all deployed Communications Suite components to Communications Suite 5. The complexity of this approach also depends on your deployment architecture. In some cases, it simply is not feasible for business reasons to upgrade an entire system at one time.

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

Table 1-5 Selective Upgrade Compared to Upgrade All

Upgrade Approach	Advantages	Disadvantages
Selective Upgrade	Minimizes number of components to upgrade	You must track the version of each component in your deployed system
Upgrade All	A consistent version for all components in your deployed system	Maximizes number of components to upgrade

The choice between selective upgrade and upgrade all is not rigid. For example, you might choose to selectively upgrade the product components on a particular computer, but wish to upgrade all shared components needed to support the selected product components.

Multi-instance Upgrades

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

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

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

Each replication technology has configuration or re-configuration procedures that might affect the overall sequence of Communications Suite 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.

Communications Suite Component Dependencies

As mentioned in “Upgrade Planning” on page 24, an upgrade plan specifies the Communications Suite components you need to upgrade and the sequence by which you need to upgrade those components. One of the important considerations in an upgrade plan is the dependencies between the various Communications Suite components in your deployed system.

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

This section provides information about Communications Suite component dependencies. The following dependency factors impact your upgrade plan:

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

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

Dependencies On Shared Components

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

Shared Component Dependency Matrix

Table 1-6 on page 31 shows the dependencies of Communications Suite 5 product components on Communications Suite shared components. The abbreviations for product components that head the columns of Table 1-6 are taken from Table 1-1 on page 18. The abbreviations for shared components are spelled out in Table 1-2 on page 19.

Two product components are not included in Table 1-6: High Availability Session Store (HADB) and Directory Preparation Tool (DPT) have been omitted because they have no dependencies on shared components.

Within the matrix of Table 1-6 hard upgrade dependencies for 2005Q1 and 2005Q4 to Communications Suite 5 upgrades are marked “H,” while soft upgrade dependencies are marked “S.” For 2004Q2 to Communications Suite 5 upgrades, all shared component dependencies are, by definition, hard upgrade dependencies; all shared components must be upgraded from 2004Q2 to Communications Suite 5.

Table 1-6 Shared Component Dependencies of Communications Suite 5 Product Components

Shared Component	CS	CX	DA	IM	MS
CAC				S	
ICU	H				H
IM SDK				H	
J2SE™	S	S	S	S	S

Table 1-6 Shared Component Dependencies of Communications Suite 5 Product Components *(Continued)*

Shared Component	CS	CX	DA	IM	MS
JSS	S		S	H	
LDAP J SDK		S	S	H	
LIBCPLUSPLUS	H				H
LIBMTMALLOC	H				H
MFWK				S	
NSPR	H		S	H	H
NSS	H		S	H	H
PATCHUTILS	H				H
PKGINSTALL	H				
SASL	H	H			H

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

Shared Component Upgrade Guidelines

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

- **2004Q2 to Communications Suite** 5 **Upgrades.** If you are performing an upgrade from 2004Q2 to Communications Suite 5, all the shared components marked as “S” or “H” in Table 1-6 for the respective product components must be upgraded.
- **2005Q1 and 2005Q4 to Communications Suite** 5 **Upgrades.** If you are upgrading all product components from 2005Q1 or 2005Q4 to Communications Suite 5, all the shared components indicated in Table 1-6 for the respective product components should be upgraded.

  Even if you are selectively upgrading product components, however, it is the recommended practice to upgrade the shared components needed by all product components on the computer; Communications Suite 5 shared components are certified to support 2005Q1 and 2005Q4 product components.

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

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

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

Also, if you plan to upgrade J2SE to J2SE 5.0, you should upgrade this shared component first. J2SE is the base component for some of the Communications Suite components.

For information on how to upgrade shared components, consult Chapter 2, "Upgrading Communications Suite Shared Components."

Dependencies On Product Components

Dependencies of one product component on another are an important determinant of the Communications Suite components you need to upgrade and the sequence by which you need to upgrade them. Dependencies on product components fall into two general categories: runtime dependencies and configuration dependencies.

- **Runtime Dependencies.** The functioning of a software system is based on the interactions between its deployed components. In upgrading any Communications Suite product component, you must take into account such dependencies. If an upgraded version of one component has a hard upgrade dependency on another component, that dependency implies that the dependent component should only be upgraded after the component upon which it depends is upgraded.
- **Configuration Dependencies.** In many cases, a Communications Suite 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 you to configure Messaging Server components. Component upgrade procedures often involve re-configuration of upgraded components. In fact, some product components' main function is to provide configuration and administrative support to other components. As a result,dependencies can have a strong impact on the sequence of upgrade procedures.

Table 1-7 shows the dependencies between the Communications Suite product components listed in Table 1-1 on page 18. Using Table 1-7, you can diagram the chain of dependencies in your upgrade set. The left column lists each product component, the middle column shows its dependencies on other product components, the third characterizes each dependency, and the last column indicates whether or not the respective components must be local.

Table 1-7 Communications Suite Product Component Dependencies

Product Component	Dependencies	Nature of Dependency	Must be Local?
Calendar Server	Directory Server	To store and enable lookup of user data	No
	Directory Preparation Tool	To prepare directory for use by Calendar Server	No
	Access Manager (optional)	To provide single sign-on	No
	Messaging Server (optional)	To provide email notifications	No
	Delegated Administrator (optional)	To provision users for calendar services	No
Communications Express	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store and enable lookup of user data, such as in address books	No
	Directory Preparation Tool	To prepare directory for use by Communications Express	No
	Access Manager or Access Manager SDK	To provide authentication and authorization services, single sign-on	Yes
	Messaging Server	To enable web-based access to messaging	No
	Calendar Server	To enable web-based access to calendaring	No

Table 1-7 Communications Suite Product Component Dependencies *(Continued)*

Product Component	Dependencies	Nature of Dependency	Must be Local?
Connector for Microsoft Outlook	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	No
	Directory Server	To store and enable lookup of user data, such as in address books	No
	Directory Preparation Tool	To prepare directory for use by Connector for Microsoft Outlook	No
	Access Manager or Access Manager SDK	To provide authentication and authorization services, single sign-on	No
	Messaging Server	To enable access to messaging	No
	Calendar Server	To enable access to calendaring	No
	Communications Express	To enable access to user's Personal Address Book	No
	Delegated Administrator (optional)	To provision users for messaging services	No
Delegated Administrator	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store user data	No
	Directory Preparation Tool	To prepare directory for use by Delegated Administrator	No
	Access Manager or Access Manager SDK	To provide API needed for user provisioning	Yes
Directory Preparation Tool	Directory Server	To provide the user/group directory that it is preparing for use by communications components	Yes
Instant Messaging	J2EE web container, one of: - Application Server - Web Server	To provide web container runtime services	Yes
	Directory Server	To store user data	No
	Access Manager (optional)	To provide single sign-on	No

Table 1-7 Communications Suite Product Component Dependencies *(Continued)*

Product Component	Dependencies	Nature of Dependency	Must be Local?
Messaging Server Store MTA MMP MEM	Directory Server	To enable lookup of user data	No
	Directory Preparation Tool	To prepare directory for use by Messaging Server	No
	Access Manager (optional)	To provide single sign-on	No
	Delegated Administrator (optional)	To provision users for messaging services	No
Sun Cluster Agents	Sun Cluster	To provide access to Sun Cluster services	Yes

General Sequencing Guidelines

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

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

NOTE The chapters in this *Upgrade Guide* are arranged according to the order you would normally upgrade Communications Suite components, as indicated by these sequencing guidelines. Instructions for upgrading Java Enterprise System components are documented in the *Java Enterprise System Upgrade Guide* (`http://docs.sun.com/doc/819-6553`).

1. **Shared Components** (See Chapter 2, “Upgrading Communications Suite Shared Components” on page 41)

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

2. **Sun Cluster software** (See the *Java Enterprise System Upgrade Guide*)

   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 Geo software** (See the *Java Enterprise System Upgrade Guide*)

   Sun Cluster Geo 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 the *Java Enterprise System Upgrade Guide*)

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

5. **Directory Proxy Server** (See the *Java Enterprise System Upgrade Guide*)

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

6. **Web Server** (See the *Java Enterprise System Upgrade Guide*)

   A number of Communications Suite 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.

7. **Message Queue** (See the *Java Enterprise System Upgrade Guide*)

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

8. **High Availability Session Store** (See the *Java Enterprise System Upgrade Guide*)

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

9. **Application Server** (See the *Java Enterprise System Upgrade Guide*)

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

10. **Service Registry** (See the *Java Enterprise System Upgrade Guide*)

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

11. **Web Proxy Server** (See the *Java Enterprise System Upgrade Guide*)

    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.

12. **Access Manager** (See the *Java Enterprise System Upgrade Guide*)

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

13. **Directory Preparation Tool** (See Chapter 3, “Directory Preparation Tool” on page 53)

    Directory Preparation Tool depends on the Directory Server schema and should therefore be run against Directory Server after Access Manager is upgraded. (An exception to this guideline exists, see Chapter 3, “Directory Preparation Tool.”) If upgrading Directory Preparation Tool, it should be upgraded before you upgrade the communications components that depend on Directory Preparation Tool to make changes in the directory: Messaging Server, Calendar Server, Communications Express, and Delegated Administrator.

14. **Messaging Server** (See Chapter 4, “Upgrading Messaging Server” on page 67)

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

15. **Calendar Server** (See Chapter 5, “Upgrading Calendar Server” on page 91)

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

16. **Communications Express** (See Chapter 6, “Upgrading Communications Express” on page 107)

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

17. **Instant Messaging** (See Chapter 7, “Upgrading Instant Messaging” on page 135)

    Instant Messaging, if upgraded, can be upgraded at almost any point after Access Manager has been upgraded. Upgrading the Instant Messaging shared components will upgrade the server on that node: the server `.jar` is part of the shared components.

18. **Portal Server** (See the *Java Enterprise System Upgrade Guide*)

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

19. **Portal Server Secure Remote Access** (See the *Java Enterprise System Upgrade Guide*)

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

20. **Delegated Administrator** (See Chapter 8, "Upgrading Delegated Administrator" on page 161)

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

Communications Suite and Solaris 10 Zones

If you are upgrading Communications Suite in an environment using Solaris 10 zones, keep the following considerations in mind:

- (Issue Number: 6527792) If you are upgrading to Communications Suite 5 from Java Enterprise System 2005Q4, you must remove the `SUNWldkx` package in the global zone before upgrading shared components.
- If you want to upgrade all installed product components to Communications Suite 5, upgrade all Communications Suite shared components in the global zone, then perform the upgrade of the desired product components in the zones where they have been installed. (Communications Suite 5 shared components are backwardly compatible.)
- If you have product components installed in a non-zones environment, and you want to add non-global zones to the environment and install product components in the new non-global zones, be sure to do so in accordance with the recommended practices described in the *Java Enterprise System Upgrade Guide*. This might mean uninstalling components in the global zone and reinstalling them in non-global zones.

For detailed information about zones, when to use them, and instructions on upgrading a deployment that uses Java Enterprise System in addition to Communications Suite components and Solaris 10 zones see the *Java Enterprise System Upgrade Guide*.

Chapter 2

Upgrading Communications Suite Shared Components

This chapter provides information on upgrading Communications Suite shared components.

Each Communications Suite product component depends on one or more locally shared libraries known as *shared* components. These shared components are installed automatically by the installer during initial product component installation. During the upgrade process, shared components are manually selected and installed, depending on the products you want to upgrade.

For Communications Suite products, you need to upgrade shared components explicitly, using the procedure described in this chapter.

The chapter contains the following sections:

Shared Component Upgrade Overview

Upgrading shared components to Communications Suite 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

When upgrading shared components, consider the following issues:

- **Operating System Issues.** Perform any operating system upgrades, as described in "Operating System Issues" on page 22. For all platforms except for Solaris 10 OS, perform operating system upgrades before you upgrade shared components.
- **Sequencing Guidelines.** Review the sequencing guidelines listed in "General Sequencing Guidelines" on page 36. Typically, shared components are upgraded first. However, you should understand the entire sequence of your upgrade to Communications Suite 5 before beginning your upgrade process.

Solaris 10 Zone Considerations

Communications Suite can be deployed in a Solaris 10 multi-zone environment, however, there are some limitations due to shared component considerations.

The major consideration impacting the use of Communications Suite in a multi-zone environment 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 Communications Suite shared components in sparse root zones means that to successfully install or to 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 and propagated to non-global zones. The Communication Suite products should be installed or upgraded in the sparse zones.

When shared components are installed in and propagate from 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 Communications Suite 5 shared components that have been propagated from the global zone.

Special care means following the practices recommended for the use of Communications Suite in a multi-zone environment (see "Communications Suite and Solaris 10 Zones" on page 39).

Special Upgrade Procedures

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

NOTE These procedures should be performed before upgrading component shared components described in "Shared Component Upgrade Procedure" on page 47.

J2SE Upgrade Procedures

Communications Suite 5 is certified for Java 2 Platform, Standard Edition (J2SE) Version 5.0 Update 6, identified here as J2SE 5.0 Update 6. (J2SE 5.0 is sometimes referred to as developer version 1.5.0).

You should not remove the previously installed J2SE Version 1.4.2 because other applications might depend on that version. Instead, make sure that your upgrade sets a symbolic link, `/usr/jdk/entsys-j2se`, to reference the Communications Suite 5 version. See the J2SE documentation for upgrade instructions.

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

When upgrading J2SE you might want to first shut down any services that depend on the currently installed J2SE. This is to avoid any problems that might arise with those services. If you do not shut down services that depend on J2SE, you should reboot your system after upgrading J2SE to 5.

Checking the J2SE Symbolic Link

You will need to check the symbolic link to the supported version of J2SE platform to ensure that Communications Suite services can find the correct J2SE runtime to use.

Check the symbolic link as follows:

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

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

Verifying the Current J2SE Version

To determine which version of J2SE your Communications Suite installation is using, run the following command, which verifies the version of J2SE referenced by the J2SE symbolic link:

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

The outputs are shown in Table 2-1 on page 44.

Table 2-1 J2SE Version Verification Outputs

Release	J2SE Version Number	J2SE Version String
Java Enterprise System 2004Q2	1.4.2 Update 5	`1.4.2_05`
Java Enterprise System 2005Q1	5.0 Update 1	`1.5.0_01`
Java Enterprise System 2005Q4	5.0 Update 4	`1.5.0_04`
Communications Suite 5	5.0 Update 6	`1.5.0_06`

Common Agent Container Upgrade Procedures

If you are upgrading the common agent container (CAC) shared component in preparation for upgrading the Monitoring Framework (MFWK) software that can be used with Instant Messaging, the following section describes the procedure for upgrading common agent container packages for standalone systems.

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

   On Solaris platforms:

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

   On the Linux platform:

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

   The output will be similar to the following:

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

   The example above lists the default values. Note any nondefault settings for use in Step 5 on page 46.

2. On the Linux platform only, back up the SNMP security files.

   ```
   mkdir /tmp/cacao_snmp
   /bin/cp /etc/opt/sun/cacao/security/snmp/* /tmp/cacao_snmp
   ```

   The security files for the Linux platform will be restored after applying Patch 120677-01.

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

   On Solaris platforms:

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

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

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

On the Linux platform:

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

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

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

4. Upgrade the common agent container.

   See "Shared Component Upgrade Procedure" on page 47.

5. Apply any custom configuration settings previously captured in Step 1 on page 45.

   On Solaris platforms, use the following commands:

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

   On the Linux platform, use the following commands:

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

   On the Linux platform only, restore the original SNMP security files you backed up in Step 2 on page 45:

   ```
   /bin/cp /tmp/cacao_snmp/* /etc/opt/sun/cacao/security/snmp/\
   rm -rf /tmp/cacao_snmp
   ```

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

   On Solaris platforms:

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

On the Linux platform:

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

The output of this command will be:

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

7. Restart common agent container services:

```
cacaoadm start
```

8. Verify the upgrade of common agent container:

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

Shared Component Upgrade Procedure

To upgrade the shared components in the Communications Suite, you will need to determine all the required shared components for your deployment and manually upgrade these components to their Communications Suite 5 versions.

NOTE Do not use the Communications Suite installer to upgrade your deployment.

Manual upgrade procedures rely on the following two upgrade technologies:

- **Patches.** Most shared components on Solaris platforms can be upgraded to Communications Suite 5 through the application of patches. Patches typically upgrade a single component or a group of related components.

  Table 2-1 on page 44 shows the upgrade patches that are available for each shared component when upgrading to Communications Suite 5 from 2005Q4, 2005Q1, or 2004Q2.

- **Replacement of Packages.** Some shared components can only be upgraded by replacing existing packages on your system with newer versions of the packages. The newer versions of shared component packages are available with your Communications Suite 5 distribution.

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

Table 2-1 on page 44 shows the shared components that use replacement of packages when upgrading to Communications Suite 5 from 2005Q4, 2005Q1, and 2004Q2.

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

1. Create an upgrade plan determining the products in the Communications Suite that you plan to upgrade.

   In addition, think about any Java Enterprise System products you need to update that are not included in the Communications Suite distribution or documentation (for example, Sun Java System Application Server). For information on upgrading those products, you will need to read the *Sun Java Enterprise System Upgrade Guide*.

   Review the earlier sections in this overview for information on developing an upgrade plan. You can also refer to "Upgrade Planning" on page 24 for additional information.

2. Determine the shared components specified in your upgrade plan. See Table 2-2 on page 49.

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

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

The full names of shared components listed in Table 2-2 are provided in "Communications Suite Shared Components" on page 19.

3. Apply the appropriate platform-specific patch required for each product in the Communications Suite that you are upgrading. These patches are available at: `http://sunsolve.sun.com/pub-cgi/show.pl?target=tous`

4. In some cases, Linux RPM packages are bundled as patches that can be downloaded from the same site. Check `http://sunsolve.sun.com/pub-cgi/show.pl?target=tous` for more information.

Table 2-2 Upgrade Technologies to Upgrade Communications Suite Shared Components from Previous Versions

Shared Component	Communications Suite Product	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
ICU	Messaging Server; Calendar Server	116103-09	114677-11	119810-02	114678-11	119811-02	`sun-icu-3.2.1-2.i386.rpm` See "Linux RPM Instructions" on page 51
IM SDK	Instant Messaging	118789-27	118789-27	118789-27	118790-27	118790-27	118791-27
J2SE™	Delegated Administrator	Install J2SE 5.0 as described in "J2SE Upgrade Procedures" on page 43.					
JSS	Instant Messaging	119209-05	119211-05	119213-06	119212-05	119214-06	Replace packages
LDAP J SDK	Instant Messaging	119725-02					Replace packages
LIBMTMALLOC	Messaging Server; Calendar Server	111308-05	115697-02	-	115698-02	-	-
LIBCPLUSPLUS	Messaging Server; Calendar Server	-	111711-16	119963-03	111713-13	119964-03	-
MFWK (Note the dependency on CAC)	Instant Messaging	119803-02			119804-02		Replace packages
		See the *Java Enterprise System Upgrade Guide* (`http://docs.sun.com/doc/819-6553`) for more information on shared component dependencies for MFWK and CAC.					
NSPR	Calendar Server, Instant Messaging, Messaging Server	119209-05	119211-05	119213-05	119212-05	119214-05	Replace packages

Table 2-2 Upgrade Technologies to Upgrade Communications Suite Shared Components from Previous Versions

Shared Component	Communications Suite Product	Solaris 8 SPARC	Solaris 9 SPARC	Solaris 10 SPARC	Solaris 9 x86	Solaris 10 x86	Linux
NSS	Messaging Server, Calendar Server, Instant Messaging	See "Additional NSS Instructions" on page 50 for details.					
PATCHUTILS	Messaging Server; Calendar Server	-	-	119254-26	-	119255-25	-
PKGINSTALL	Calendar Server	-	113713-21	-	114568-20	-	-
SASL	Messaging Server, Calendar Server	115328-02	115342-02	119345-01	115343-02	119346-01	`sun-sasl-2.19-4.i386.rpm` See "Linux RPM Instructions" on page 51.

Additional NSS Instructions

You only need to apply the NSS patches listed in Table 2-3 if the package version of `SUNWtls` is the following:

Table 2-3 SUNWtls Version for NSS Patches

Operating System	SUNWtls Version needing NSS Patch	NSS Patch
Solaris 8 Sparc	3.3.2,REV=2002.09.18.12.49	119209-10
Solaris 9 Sparc	3.3.2,REV=2002.09.18.12.49	119211-10
Solaris 10 Sparc	3.9.5,REV=2005.01.14.17.27	119213-10
Solaris 9 x86	3.3.3,REV=2003.01.09.17.07	119212-10
Solaris 10 x86	3.9.5,REV=2005.01.14.19.03	119214-10
Linux	Not Applicable	121656-10

To check the package version of SUNWtls, use the following command:

```
pkgparam SUNWtls VERSION
```

Linux RPM Instructions

Some Linux rpms are not created into patches. Therefore, you need to manually apply the rpm.

The following rpms should be used to upgrade the ICU and SASL shared components:

```
ICU:  sun-icu-3.2.1-2.i386.rpm
SASL: sun-sasl-2.19-4.i386.rpm
```

To upgrade existing Linux shared component rpms to the following shared component rpms, cd into the following directory:

`Product/shared_components/Packages`

and then use the following command to apply the new rpms:

```
rpm -F sun-icu-3.2.1-2.i386.rpm
rpm -F sun-sasl-2.19-4.i386.rpm
```

If you are not upgrading Application Server to version 8.x, you also need to complete the following steps:

1. Open *AppServer8-base*`/config/asenv.conf`.
2. Change the AS_ANT_LIB value from:

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

   to:

   `AS_ANT_LIB="/opt/sun/share/lib"`
3. Restart Application Server.

Chapter 3

Directory Preparation Tool

This chapter describes how to upgrade Directory Preparation Tool to Version 6.4.

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

NOTE File locations in this chapter are specified with respect to a directory path referred to as *DirPrepTool-base*. At least part of this path might have been specified as an installation directory when Directory Preparation Tool was initially installed. If not, the version installer assigned a default value.

The default value of *DirPrepTool-base* depends on operating system platform, as shown in the following table.

Path Name	Solaris OS	Linux OS
DirPrepTool-base	`/opt/SUNWcomds`	`/opt/sun/comms/dssetup`

Overview of Directory Preparation Tool Upgrades

This section describes the following general aspects of Directory Preparation Tool that impact upgrading to Directory Preparation Tool 6.4:

About the Directory Preparation Tool

The Directory Preparation Tool represents a minor upgrade with respect to the previous release, mostly to add additional support for Directory Server.

For details, see the appropriate release notes.

Directory Preparation Tool Upgrade Roadmap

Table 3-1 shows the supported Directory Preparation Tool upgrade paths to Version 6.4. The table applies to both Solaris and Linux operating systems.

Table 3-1 Upgrade Paths to Sun Java System Directory Preparation Tool Version 6.4

Release	Directory Preparation Tool Version	General Approach	Re-configuration Required
Java Enterprise System 2005Q4	Sun Java System Directory Preparation Tool 6.3 2005Q4	Direct upgrade: Perform by applying patches.	Prepare Directory Server for 6.4 communications components
Java Enterprise System 2005Q1	Sun Java System Directory Preparation Tool 6.2 2005Q1	Direct upgrade: Perform by applying patches.	Prepare Directory Server for 6.4 communications components
Java Enterprise System 2004Q2	`comm_dssetup.pl` script Version 6.1 Revision 0.2 (bundled with Messaging Server and Calendar Server)	Direct upgrade: Perform by applying genesis patch followed by an upgrade patch.	Prepare Directory Server for 6.4 communications components

Table 3-1 Upgrade Paths to Sun Java System Directory Preparation Tool Version 6.4 *(Continued)*

Release	Directory Preparation Tool Version	General Approach	Re-configuration Required
Java Enterprise System 2003Q4	`comm_dssetup.pl` script (bundled with Messaging Server and Calendar Server)	Direct upgrade not certified: But can perform by applying genesis patch followed by an upgrade patch.	Prepare Directory Server for 6.4 communications components
5.2 and earlier	`ims_dssetup.pl` script (bundled with Messaging Server)	No direct upgrade.	

Directory Preparation Tool Data

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

Table 3-2 Directory Preparation Tool Data Usage

Type of Data	Location	Usage
Directory Server schema	Directory Server	Prepare Directory Server for 6.4 communications components: modifies schema, creates new entries and creates indexes

Directory Preparation Tool Upgrade Strategy

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

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

Compatibility Issues

This release of the Directory Preparation Tool does not introduce any interface changes and is backwardly compatible with earlier versions.

Directory Preparation Tool Dependencies

Directory Preparation Tool has no dependencies on other components other than Directory Server. Directory Preparation Tool is used to configure Directory Server for use with communications components.

Upgrading Directory Preparation Tool from 6.3 (2005Q4)

This section includes information about upgrading Directory Preparation Tool from Version 6.3 (2005Q4). The section covers the following topics:

Introduction

When upgrading the Directory Preparation Tool from 6.3 (2005Q4) to 6.4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the 6.3 (2005Q4) version. Directory Preparation Tool is then used to modify Directory Server as required to support the most recent versions of Messaging Server, Calendar Server, Communications Express, and Delegated Administrator components.
- **Upgrade Dependencies.** Directory Preparation Tool has no dependencies on shared components and is compatible with Sun Java System 2005Q4 Directory Server. Upgrade of Directory Server is therefore optional with respect to upgrade of Directory Preparation Tool to Version 6.4.
- **Backward Compatibility.** Version 6.3 (2005Q4) Directory Preparation Tool is backwardly compatible with the 6.4 version.
- **Upgrade Rollback.** Rollback of the 6.4 upgrade to 6.3 (2005Q4) is achieved by removing the patches applied during the upgrade. The 6.3 (2005Q4) Directory Preparation Tool can then be run against Directory Server to back out changes made by the 6.4 version.

- **Platform Issues.** The general approach for upgrading Directory Preparation Tool is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures, and you normally cannot roll back patches on the Linux platform.

Version 6.3 (2005Q4) Directory Preparation Tool Upgrade

This section describes how to perform an upgrade of Directory Preparation Tool from 6.3 (2005Q4) to Version 6.4 on both 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

Before you upgrade Directory Preparation Tool you should perform the tasks described below.

Verify Current Version Information

You can verify the version of Directory Preparation Tool last run against Directory Server by checking attribute values of the `cn=CommServers,o=comms-config` entry written by the tool:

```
./ldapsearch -D “cn=Directory Manager” -w password
             -b cn=CommServers,o=comms-config cn=”CommServers”
             sunkeyvalue
```

The entry has two attributes that specify the current version:

- `dssetup_ver`=*version* (for example, `6.3`)
- `dssetup_rev`=*revision* (for example, `2.01`)

Table 3-3 Directory Preparation Tool Version Verification Outputs

Release	dssetup_ver	dssetup_rev
Java Enterprise System 2003Q4	`6.0`	`0.004`
Java Enterprise System 2004Q2	`6.1`	`0.2`
Java Enterprise System 2005Q1	`6.3`	`1.0`
Java Enterprise System 2005Q4	`6.3`	`2.03`
Communications Suite 5	`6.4`	`xx`

The tool will write a message to console only if the version of Directory Preparation Tool being run is the same or earlier than the version that was previously run. See the upgrade procedures, Step 5 on page 60 (Solaris) or Step 5 on page 61 (Linux), for how to run the tool.

Upgrade Directory Preparation Tool Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 5. However, the upgrade of Directory Preparation Tool does not depend upon any other Communications Suite component.

Back Up Directory Data

The Directory Preparation Tool upgrade from 6.3 (2005Q4) to 6.4 modifies Directory Server data in only minor ways. However, as a safety measure, it is a good idea to back up Directory Server before upgrading the Directory Preparation Tool and using it to modify Directory Server.

Obtain Required Configuration Information and Passwords

Directory Preparation Tool upgrade requires you to know the superuser password. The tool remembers parameter values used in the previous run and supplies them as defaults when run the next time.

Upgrading 6.3 (2005Q4) Directory Preparation Tool (Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Directory Preparation Tool software to Version 6.4 takes into account the following considerations:

- Version 6.3 (2005Q4) Directory Preparation Tool was installed with Directory Server and resides on any computer hosting Directory Server.
- The upgrade of Directory Preparation Tool must be performed on the computer hosting every Directory Server instance being used by Messaging Server, Calendar Server, Communications Express, or Delegated Administrator components.
- The 6.4 Directory Preparation Tool upgrade patch for Solaris OS are shown in the following table:

Table 3-4 Patches[1] to Upgrade Directory Preparation Tool on Solaris

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Directory Preparation Tool (DSSETUP)	118245-13	118246-13

1. Patch revision numbers are the minimum required for upgrade to Version 6.4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Obtain the required patches, based on Table 3-4.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Apply the appropriate Directory Preparation Tool patches in Table 3-4.

   `patchadd` *patch_ID*

4. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 3.

5. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      ```
      perl comm_dssetup.pl
      ```

      Provide the parameters requested by the script.

Upgrading 6.3 (2005Q4) Directory Preparation Tool (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Directory Preparation Tool software to Version 6.4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 59), except that the Linux 6.4 upgrade patches differ from the Solaris patches.

The 6.4 Directory Preparation Tool upgrade patch for Linux OS is shown in the following table:

Table 3-5 Patches[1] to Upgrade Directory Preparation Tool on Linux

Description	Patch ID and RPM names
Directory Preparation Tool (DSSETUP)	118247-13 • `sun-comms-dssetup-6.3-3.xx.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Version 6.4. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to the Directory Preparation Tool installed image on the computer where Directory Server resides.

CAUTION An upgrade from Version 6.3 (2005Q4) to Version 6.4 on Linux cannot be rolled back.

1. Obtain the required patch using the patch number and RPM name from Table 3-5. Use this information to obtain the version numbers for the RPM.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Apply the RPMs for Directory Preparation Tool in Table 3-5.

   ```
   rpm -Uvh sun-comms-dssetup-6.3-3.xx.i386.rpm
   ```

4. Confirm that the upgrade was successful:

   ```
   rpm -q sun-comms-dssetup
   ```

   The new version number of the RPM should be returned.

5. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      ```
      perl comm_dssetup.pl
      ```

      Provide the parameters requested by the script.

Verifying the Upgrade

You can verify successful upgrade of Directory Preparation Tool and extension of directory schema by checking the log file created when running the script. The log file is located at:

`/var/tmp/dssetup_`*yyyymmddhhmmss*`.log`

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in “Upgrade Procedure (Solaris)” on page 59 and “Upgrade Procedure (Linux)” on page 60.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Directory Preparation Tool followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to 6.4 of Directory Preparation Tool reverses the procedure for upgrading to 6.4. However, among the changes made by Directory Preparation Tool are modifications to Directory Server schema. These changes are not backed out by the rollback procedure described below, however the schema changes are backwardly compatible.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   `su -`

2. Remove the patches in Table 3-4 on page 59.

   `patchrm` *patch_ID*

3. Run the rolled-back Directory Preparation Tool against Directory Server.

   Directory Server modifications, except for indexes and schema changes, are restored to their previous states. There is no negative impact to the indexes or schema extensions; the latter are backwardly compatible.

Upgrading Directory Preparation Tool from Version 6.3 (2005Q1)

The procedure for upgrading Version 6.3 (2005Q1) Directory Preparation Tool to 6.4 is the same as that for upgrading 6.3 (2005Q4) Directory Preparation Tool to 6.4.

To upgrade 6.3 (2005Q1) Directory Preparation Tool to 6.4, use the instructions in "Upgrading Directory Preparation Tool from 6.3 (2005Q4)" on page 56, except substitute 6.3 (2005Q1) wherever 6.3 (2005Q4) is referenced.

Upgrading Directory Preparation Tool from Version 6.1 (2004Q2)

The procedure for upgrading Version 6.1 (2004Q2) Directory Preparation Tool to 6.4 is similar to that for upgrading 6.3 (2005Q1) Directory Preparation Tool to 6.4, with the following exception:

In Version 6.1 (2004Q2), Directory Preparation Tool was bundled with Messaging Server and Calendar Server and not installed as a separate package. Hence no Directory Preparation Tool installed packages or RPMs reside on the computer hosting Directory Server. For this reason, to upgrade from 6.1 (2004Q2) to 6.4, you have to install Directory Preparation Tool packages:

- For Solaris platforms, the DPT packages are installed as genesis patches, which contain the full Directory Preparation Tool software. You then apply patches to upgrade to 6.4.
- For Linux platforms, the 6.4 packages are directly installed.

Upgrades from 6.1 (2004Q2) Directory Preparation Tool to 6.4 is similar to that described in "Upgrading Directory Preparation Tool from 6.3 (2005Q4)" on page 56. The pre-upgrade and post-upgrade considerations are the same, except you substitute 6.1 (2004Q2) wherever 6.3 (2005Q1) is referenced. The specific upgrade procedures, however, are described in the following sections.

Version 6.1 (2004Q2) Upgrade Procedure (Solaris)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Obtain the required genesis patch, based on the following table:

Table 3-6 Genesis Patches[1] to Upgrade Directory Preparation Tool on Solaris

Description	SPARC Solaris 9,& 10	X86 Solaris 9 & 10
Directory Preparation Tool (DSSETUP)	118242-03	118243-03

1. Patch revision numbers are the minimum required for upgrade to Version 6.4. If newer revisions become available, use the newer ones instead of those shown in the table.

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

2. Log in as root or become superuser.

```
su -
```

3. Apply the Directory Preparation Tool genesis patch in Table 3-6.

   `patchadd` *patch_ID*

4. Obtain the required upgrade patch, based on Table 3-4.

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

5. Apply the appropriate Directory Preparation Tool upgrade patch in Table 3-4.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrade was successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      `perl comm_dssetup.pl`

      Provide the parameters requested by the script.

Version 6.1 (2004Q2) Upgrade Procedure (Linux)

The procedure documented below applies to the Directory Preparation Tool installed on the computer where Directory Server resides.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Obtain the required upgrade patch using the patch number and RPM name from Table 3-5 on page 60.

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

3. Apply the upgrade RPMs for Directory Preparation Tool in Table 3-5 on page 60.

   ```
   rpm -Uvh sun-comms-dssetup-6.3-2.5.i386.rpm
   ```

4. Confirm that the upgrade was successful:

   ```
   rpm -q sun-comms-dssetup
   ```

   The new version number of the RPM should be returned.

5. Run the Directory Preparation Tool against Directory Server.

   a. Confirm that Directory Server is running.

   b. Change directory to the location of the Directory Preparation Tool

      `cd` *DirPrepTool-base*`/sbin`

   c. Run the Directory Preparation Tool (`comm_dssetup.pl` perl script).

      ```
      perl comm_dssetup.pl
      ```

      Provide the parameters requested by the script.

Chapter 4

Upgrading Messaging Server

This chapter describes how to upgrade previous versions of Messaging Server to Sun Java System Messaging Server 6.3.

The chapter provides an overview of upgrade considerations for the different supported upgrade paths 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 *msg-svr-base*. At least part of this path might have been specified as an installation directory when Messaging Server was initially installed. If not, the installer assigned a default value.

The default value of *msg-svr-base* depends on operating system platform, as shown in the following table.

Path Name	Solaris OS	Linux OS
msg-svr-base	`/opt/SUNWmsgsr`	`/opt/sun/messaging`

Overview of Messaging Server Upgrades

This section describes the following general aspects of Messaging Server that impact upgrade:

- About Messaging Server 6.3
- Messaging Server Upgrade Roadmap
- Messaging Server Data
- Messaging Server Upgrade Strategy

About Messaging Server 6.3

Messaging Server has two private interface changes that affect upgrade:

- Messaging Server 6.3 configuration data is no longer stored in a Directory Server instance called the configuration directory. This data is now stored as configuration variables in a local configuration file. Configuration information is no longer managed through the Administration Server and its administrative console interface.
- The Messenger Express Multiplexor (MEM) has been modified to eliminate its multiplexor function and to function only as a webmail server that accesses the Message Store. This webmail server interacts with the back-end Message Store using the IMAP protocol, which requires that new configuration parameters be set after upgrade. It is now called the Webmail Server.

Messaging Server Upgrade Roadmap

Table 4-1 shows the supported Messaging Server upgrade paths on both Solaris and Linux operating systems.

Table 4-1 Upgrade Paths to Sun Java System Messaging Server 6.3

Release	Messaging Server Version	General Approach	Re-configuration Required
Java Enterprise System 2005Q4 (6.2p3)	Sun Java System Messaging Server 6.2 2005Q4	Direct upgrade: Performed by applying patches.	Configuration data migrated to 6.3 configuration files and updated.

Table 4-1 Upgrade Paths to Sun Java System Messaging Server 6.3 *(Continued)*

Release	Messaging Server Version	General Approach	Re-configuration Required
Java Enterprise System 2005Q1 (6.2)	Sun Java System Messaging Server 6.2 2005Q1	Direct upgrade: Performed by applying patches.	Configuration data migrated to 6.3 configuration files and updated.
Java Enterprise System 2004Q2 (6.1)	Sun Java System Messaging Server 6.1 2004Q2	Direct upgrade: Performed by applying patches. Some additional manual steps on Linux	Configuration data migrated to 6.3 configuration files and updated.
Java Enterprise System 2003Q4 (6.0) Solaris OS only	Sun ONE Messaging Server 6.0 (2003Q4)	Direct upgrade not certified: Performed by applying patches.	Configuration data migrated to 6.3 configuration files and updated.
Pre-dates Communications Suite releases (5.2 and earlier)	Sun ONE Messaging Server 5.2	Direct upgrade not certified: But you can upgrade to 6.3 using procedures in the *Java Enterprise System 2005Q1 Upgrade and Migration Guide* (http://docs.sun.com/doc/819-0062).	Configuration data migrated to 6.3 configuration files and updated.

Messaging Server Data

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

Table 4-2 Messaging Server Data Usage

Type of Data	Location	Usage
Configuration data	Local configuration directory: *msg-svr-base*/config/msg.conf and many other configuration files for configuring Message Store, MTA, MMP, Webmail Server (formerly known as MEM)	Configuration of Messaging Server components

Table 4-2 Messaging Server Data Usage

Type of Data	Location	Usage
	Prior to 6.3, some configuration data was stored in the Directory Server configuration directory	
User data	Directory Server user/group directory	Storing user attributes needed to support messaging for end users
Dynamic application data	Message store: *msg-svr-base*/data/store	Store email messages, message transfer queues, and related information on behalf of users
Directory schema	Directory Server (2004Q2, 2005Q1, and 2005Q4) /var/opt/mps/serverroot Directory Server 6.0 Solaris: var/opt/sun/dsins1 Linux: /var/opt/sun/directory-server	For user attributes needed to support end users

Messaging Server Upgrade Strategy

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

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

Compatibility Issues

Messaging Server 6.3 does not introduce any changes in public interfaces. The logically distinct configurations of Messaging Server (Message Store, MTA, and MMP) have the same public interfaces, and are therefore backwardly compatible with 2005Q4, however the protocol between the Message Store and MEM (now known as Webmail Server) has changed in this release, impacting the upgrade procedure. See "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81.

Messaging Server Dependencies

Messaging Server dependencies on other Communications Suite components and on Java Enterprise System components can impact the procedure for upgrading and re-configuring Messaging Server software. Changes in Messaging Server interfaces or functions, for example, could require an upgraded version of components upon which Messaging Server depends. The need to upgrade such components depends upon the specific upgrade path.

Messaging Server has dependencies on the following Java Enterprise System components:

- **Shared components.** Messaging Server has dependencies on specific Communications Suite shared components (see Table 1-6 on page 31).
- **Directory Server.** Message stores user data needed for Messaging in Directory Server. As a result, Messaging Server upgrades might require extensions of directory schema. Directory Server is no longer needed to store configuration data.
- **Directory Preparation Tool.** Messaging Server uses the Directory Preparation Tool to prepare Directory Server to support Messaging Server functions.
- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Messaging Server can be configured to use Access Manager single sign-on capability.
- **Message Queue (optional).** Message Queue is used by Messaging Server for asynchronous notifications.

Deployment Architectures

In most deployment architectures involving Messaging Server, the logically distinct configurations of Messaging Server (Message Store, MTA, MMP, and MEM components) are deployed on different computers and often replicated for high availability and/or scalability.

These architectures can create upgrade challenges. In some cases the Messaging Server components need to be upgraded in a particular sequence, for example, because of changes in internal protocols. In many cases the upgrade needs to be performed as a rolling upgrade that does not interrupt service.

For documentation regarding some of these issues and how to resolve them, see “Multiple Instance Upgrades” on page 83.

Upgrading Messaging Server from Messaging Server 6.2p3 (Java Enterprise System 2005Q4)

This section includes information about upgrading Messaging Server 6.2p3 (Java Enterprise System 2005Q4) to Messaging Server 6.3. The section covers the following topics:

Introduction

When upgrading Messaging Server 6.2p3 (Java Enterprise System 2005Q4) to Messaging Server 6.3, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the 2005Q4 version. Re-configuration is achieved by running two data configuration utilities and by importing configuration data into Directory Server.
- **Upgrade Dependencies.** While Messaging Server has dependencies on a number of Communications Suite shared components (see Table 1-6 on page 31), Messaging Server requires that NSS/ NSPR be upgraded. Upgrade of J2SE is optional with respect to upgrade of Messaging Server.

  In addition, Messaging Server is dependent upon Directory Server and optionally dependent on Access Manager, as described in "Messaging Server Dependencies" on page 70. These are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Messaging Server.

  However, Messaging Server has a hard upgrade dependency on Directory Preparation Tool; the Directory Preparation Tool is required to prepare Directory Server for messaging operations. See Chapter 3, "Directory Preparation Tool."
- **Backward Compatibility.** Messaging Server 6.3 is backwardly compatible with the 6.2p3 (Java Enterprise System 2005Q4) version.

- **Upgrade Rollback.** Rollback of the upgrade of Messaging Server to 6.2p3 (Java Enterprise System 2005Q4) is achieved by first removing the changes made to Directory Server, removing changes to local configuration files, and removing the patches applied during the upgrade.
- **Platform Issues.** The general approach for upgrading Messaging Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Messaging Server 6.2p3 (Java Enterprise System 2005Q4) Upgrade

This section describes how to perform an upgrade of Messaging Server 6.2p3 (Java Enterprise System 2005Q4) to version 6.3 on both 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

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

Verify Current Version Information

You can verify the current version of Messaging Server by entering the following command:

msg-svr-base`/sbin/imsimta version`

Table 4-3 Messaging Server Version Verification Outputs

Messaging Server Release	Messaging Server Version Number
2004Q2	`6.1`
2005Q1	`6.2`
2005Q4	`6.2p3 (6.2-3.04)`
As part of Communications Suite 5	`6.3-0.08`

Upgrade Messaging Server Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 5. However, Messaging Server has hard upgrade dependencies on a number of shared component and on Directory Preparation Tool. Upgrading of other Communications Suite 5 product components upon which Messaging Server depends is therefore optional.

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

1. **Shared Components.** If shared components have not yet been upgraded, you should do so by following the instructions in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Access Manager.** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.
4. **Message Queue.** Instructions for upgrading Message Queue are provided in the *Java Enterprise System Upgrade Guide*.
5. **Directory Preparation Tool.** Instructions for upgrading Directory Preparation Tool are provided in the Chapter 3, "Directory Preparation Tool."

Back Up Messaging Server Data

The Messaging Server upgrade from 6.2p3 to 6.3 requires re-configuration of Messaging Server in a number of local configuration files and writing some data to Directory Server. The local changes can be rolled back, but it is a good idea to back up the Directory Server in case you want to roll back the upgrade at a future point.

Obtain Required Configuration Information and Passwords

Messaging Server upgrade requires knowing the following information:

- Superuser password
- Directory Manager DN and password

Upgrading Messaging Server 6.2p3 (Solaris)

This section discusses considerations that impact the upgrade procedure for Messaging Server 6.2p3 followed by a description of the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Messaging Server software to Communications Suite 5 takes into account the following considerations:

- All logically distinct configurations of Messaging Server (Message Store, MTA, MMP, and Webmail Server (MEM) components) that correspond to the same installed Messaging Server image, should be upgraded together. All such components should be shut down before patches are applied to the installed image.
- Because of changes in protocol between the Message Store and Webmail Server (MEM) components, the Message Store should be upgraded and re-configured before upgrading the Webmail Server (MEM) component that interact with the Store. For more detailed information, see “Multiple Front-end and Back-end Components” on page 84 and
- The Messaging Server upgrade patches for Solaris OS are shown in the following table. The core patch migrates configuration data from the Directory Server configuration directory to local configuration files.

Table 4-4 Patches[1] to Upgrade Messaging Server 6.2p3 on Solaris

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Messaging Server core software with S/MIME	120228-17	120229-17
Messaging Server localization	117784 -17	117785 -17

1. Patch revision numbers are the minimum required for upgrade to Communications Suite 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 all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

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

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

2. Log in as root or become superuser.

   `su -`

3. Stop all running Messaging Server components.

   msg-svr-base`/sbin/stop-msg`

4. If you have not already done so, upgrade shared components.

   "Upgrade Messaging Server Dependencies" on page 74.

5. Apply the appropriate Messaging Server core and, if needed, localization patches in Table 4-4, in that order.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrades were successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Migrate configuration data from existing configuration files and the Directory Server configuration directory to 6.3 configuration files.

   a. Create configuration files corresponding to the core and, if needed, localization patches.

      ```
      cd msg-svr-base/sbin
      ./patch-config msg-svr-base/install/patch/patch_ID
      ```

      This command backs up existing configuration files. Then it merges configuration parameter values in these files with 6.3 template configuration files to create new 6.3 configuration files. In addition, configuration that was previously stored in the Directory Server configuration directory is migrated to the new 6.3 configuration files. You should examine these new files for possible conflicts, as described in the Special Installation Instructions section of the core patch `readme` file.

      This command also generates the following `ldif` file (LDAP directory import files):

      msg-svr-base`/lib/patch/ugdir_diff.ldif`

   b. Install the 6.3 configuration files corresponding to the core and, if needed, localization patches, making them the active configuration.

      ```
      ./install-newconfig -f msg-svr-base/install/patch/patch_ID
      ```

      This command installs the new 6.3 configuration files in their correct locations. The `-f` flag avoids having to confirm the operation for each file.

   c. Import the new configuration data generated in Step a on page 77 into the Directory Server instance being used by Messaging Server.

      Change to the Directory Server instance and import the `ldif` file using the `ldapmodify` command:

      ```
      cd msg-svr-base/lib
      ./ldapmodify -D bind_dn -w password -c -h hostName -p port
          -e patch/ugdir_diff.rej -f patch/ugdir_diff.ldif
      ```

NOTE If the `ldapmodify` command fails on the Solaris platform, first set the library path appropriately for your shell to:

```
LD_LIBRARY_PATH=msg-svr-base/lib
```

d. If the IMAP proxy username and password are different from the username and password of the store the webmail server is accessing via IMAP, you need to set `LD_LIBRARY_PATH` as follows.

```
LD_LIBRARY_PATH= msg-svr-base/sbin/configutil -o
     local.service.proxy.admin -v admin_ID

LD_LIBRARY_PATH= msg-svr-base/sbin/configutil -o
     local.service.proxy.adminpass -v password
```

The default value of `local.service.proxy.admin` is `admin`. The default value of `local.service.proxy.adminpass` is the password you entered at configure time.

8. Increase the number of file descriptors by setting `ulimit` as follows:

```
ulimit -n <number of file descriptors>
```

For example:

```
ulimit -n 100000
```

Configure ulimit with a value no less than 12851.

9. Restart the Messaging Server components that were stopped in Step 3.

msg-svr-base`/sbin/stop-msg`

msg-svr-base`/sbin/start-msg`

Upgrading Messaging Server 6.2p3 (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Messaging Server software on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 75), except that the Linux upgrade patches differ from the Solaris patches.

The Messaging Server upgrade patches for Linux OS are shown in the following table:

Table 4-5 Patches[1] to Upgrade Messaging Server on Linux

Description	Patch ID and RPM names
Messaging Server core software with S/MIME	120230-17 • `sun-messaging-server-6.3-13.9.i386.rpm`

Table 4-5 Patches[1] to Upgrade Messaging Server on Linux *(Continued)*

Description	Patch ID and RPM names
Messaging Server localization	117786-17 • `sun-messaging-<`*locale*`>-6.3-1.7.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Communications Suite 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 all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

CAUTION An upgrade from Messaging Server 6.2p3 to Messaging Serevr 6.3 on Linux cannot be rolled back.

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

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop all running Messaging Server components.

   msg-svr-base`/sbin/stop-msg`

4. If you have not already done so, upgrade shared components.

   "Upgrade Messaging Server Dependencies" on page 74.

5. Apply the RPMs for Messaging Server core and, if needed, localization patches in Table 4-5, in that order.

   For example:

   ```
   rpm -Fvh sun-messaging-server-6.3-13.9.i386.rpm
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -q sun-messaging-server
   ```

   The new version number of the RPM should be returned.

7. Migrate configuration data from existing configuration files and the Directory Server configuration directory to 6.3 configuration files.

   a. Create configuration files corresponding to the core and, if needed, localization patches.

      ```
      cd msg-svr-base/sbin
      ./patch-config msg-svr-base/install/patch/patch_ID
      ```

      This command backs up existing configuration files. Then it merges configuration parameter values in these files with 6.3 template configuration files to create new configuration files. In addition, configuration previously stored in the Directory Server configuration directory is migrated to the new 6.3 configuration files. You should examine these new files for possible conflicts, as described in the Special Installation Instructions section of the patch 12030 `README` file.

      This command also generates the following `ldif` files (LDAP directory import files):

      msg-svr-base`/lib/patch/ugdir_diff.ldif`

   b. Install the 6.3 configuration files corresponding to the core and, if needed, localization patches, making them the active configuration.

      ```
      ./install-newconfig msg-svr-base/install/patch/patch_ID
      ```

      This command installs the new configuration files in their correct locations.

   c. Import the new configuration data generated in Step a into the Directory Server instance being used by Messaging Server.

      Change to the Directory Server instance and import the `ldif` file using the `ldapmodify` command:

      ```
      cd msg-svr-base/lib
      ./ldapmodify -D bind_dn -w password -c -h hostName -p port
          -e patch/ugdir_diff.rej -f patch/ugdir_diff.ldif
      ```

NOTE If the `ldapmodify` command fails on the Linux platform, first set the library path appropriately for your shell to:

```
LD_LIBRARY_PATH=msg-svr-base/lib
```

8. Change the number of file descriptors by setting `ulimit` as follows:

   `ulimit -n <`*number of file descriptors*`>`

   For example:

   ```
   ulimit -n 100000
   ```

9. Restart the Messaging Server components that were stopped in Step 3.

   msg-svr-base`/sbin/start-msg`

Verifying the Messaging Server Upgrade to Version 6.3

You can verify the upgrade of Messaging Server to version 6.3 by entering the following command:

msg-svr-base`/sbin/imsimta version`

You can also check the banner displayed when starting up Messaging Server components.

See Table 4-3 on page 74 for output values.

Post-Upgrade Tasks for Messaging Server Version 6.3

If you used MEM (Messenger Express Multiplexor) in previous releases, the new Webmail Server now interacts with the Message Store using the IMAP protocol.

CAUTION If you are using Communications Express, the Webmail Server, and back-end Message Stores, the following deployment scenarios are supported:

- Version 6.3 Communications Express with Version 6.3 Webmail Server and Version 6.3 back-end Message Store(s)
- Version 6.2 Patch 3 (2005Q4) Communications Express with Version 6.2 Patch 3 (2005Q4) MEM with Version 6.3 back-end Message Store(s)

The following upgrade scenario is not supported:

- Version 6.2 Patch 3 (2005Q4) Communications Express with Version 6.3 Webmail Server with Version 6.2 Patch 3 (2005Q4) back-end Message Store(s)

When upgrading from Version 6.2 Patch 3 (2005Q4), you must upgrade Message back-end Store(s) first and make sure that IMAP is enabled before upgrading and re-configuring Webmail Server that accesses the Message Store. In addition, you must make sure that the Webmail Server and Communications Express are upgraded together and are of the same version.

In other words, you must perform the following steps:

1. After upgrading the Message Store, enable the IMAP protocol.

   Perform the following `configutil` command:

   msg-svr-base`/sbin/configutil -o service.imap.enable -v yes`

2. After upgrading MEM to Webmail Server, configure it to access the Message Store.

   You need to specify the IMAP admin user and password. Webmail Server can be configured to interact with more than one Message Store by setting respective `configutil` variables, each specifying a target host name, as indicated below. If there is only one Message Store, omit the *.hostname*.

   msg-svr-base`/sbin/configutil`
   `-o service.http.allowadminproxy.`*hostname* `-v yes`

   msg-svr-base`/sbin/configutil`
   `-o local.service.http.proxy.admin.`*hostname*
   `-v` *webmailProxyAdmin user ID*

   msg-svr-base`/sbin/configutil`
   `-o local.service.http.proxy.adminpass.`*hostname*
   `-v` *webmailProxyAdmin password*

3. After all components have been upgraded for a Message Store, you can disable and shut down the `mshttpd` process, as follows:

   msg-svr-base`/sbin/configutil -o service.http.enable -v no`
   msg-svr-base`/sbin/stop-msg http`

   In a single-host environment, you can't shut down `mshttpd`, otherwise Webmail Server will also be shut down.

   However, in a distributed environment where you had MEM on front ends and `mshttpd` on the back-end prior to upgrade, you will now need to shut down `mshttpd` on the back-end server (co-located with the store) after upgrade and reconfiguration.

4. After all components have been migrated to 6.3, the Directory Server configuration directory server is no longer accessed. It can be decommissioned.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Messaging Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to a previous version of Messaging Server is for the most part the reverse of the procedure for upgrading the newest version. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop all running Messaging Server components.

   msg-svr-base`/sbin/stop-msg`

3. Roll back the changes made to the Directory Server user/group directory being used by Messaging Server.

   Replace the directory with the pre-upgrade directory that you backed up before beginning the upgrade procedure (see "Back Up Messaging Server Data" on page 74).

4. Roll back the re-configuration performed in Step 7 on page 77.

   ```
   cd msg-svr-base/sbin
   ./uninstall-newconfig -f msg-svr-base/install/patch/patch_ID
   ```

5. Remove the core and, if applied, localization patches in Table 4-4 on page 75.

   ```
   patchrm patch_ID
   ```

6. Restart the Messaging Server components that were stopped in Step 2.

   msg-svr-base`/sbin/start-msg`

Multiple Instance Upgrades

In most deployment architectures involving Messaging Server, the logically distinct configurations of Messaging Server (Message Store, MTA, MMP, and Webmail Server) are deployed on different computers and often replicated for high availability and/or scalability.

For example, you might have MTA or MMP components running on multiple computers with a load balancer to distribute the load. You might also have the Message Store component running in a Sun Cluster environment to provide high availability.

Multiple Front-end and Back-end Components

In deployment architectures in which various Messaging Server subcomponents (Message Store, MTA, MMP, Webmail Server) are deployed on different computers, it is generally best to upgrade components beginning in the back-end tier (Message Store) and working toward the front-end tier (MTA, MMP, and Webmail Server).

This approach is particularly important when upgrading MEM components to the Webmail Server because the protocol to access Message Store has changed. In this case Message Store needs to be upgraded and the IMAP protocol enabled (see "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81) before upgrading to Webmail Server.

If both a Message Store and Webmail Server are configured for the same computer, be sure to upgrade any remote Message Stores accessed by the Webmail Server before upgrading Messaging Server on the local computer.

You perform the upgrade of each instance as described in "Messaging Server 6.2p3 (Java Enterprise System 2005Q4) Upgrade" on page 73.

In the case of load-balanced instances of Messaging Server, you can perform a rolling upgrade in which you upgrade the Messaging Server instances sequentially without interrupting service. You upgrade each instance of Messaging Server while the others remain running.

High Availability Clusters

In the case of Message Store running in a cluster environment, the active Messaging Server instance nodes and the failover nodes share the same configuration and user data, located on a shared file system. Generally speaking, the Messaging Server image is also installed on the shared file system. Depending on your cluster configuration you can minimize upgrade downtime by performing a rolling patch upgrade, as described below.

This approach is possible because the 6.3 binaries are compatible with the 6.2p3 configuration files and other data. However it is important to note that once you have started using 6.3 with a configuration, you should avoid reverting back to the 6.2p3 binaries. The reason is that some 6.3 generated data (e.g. queue files) might not be compatible with 6.2p3.

Asymmetric HA

The basic asymmetric or "hot standby" high availability model consists of two clustered host machines or "nodes." A logical IP address and associated host name are designated to both nodes.

In this model, only one node is active at any given time; the backup or hot standby node remains idle most of the time. A single shared disk array between both nodes is configured and is mastered by the active or “primary” node. The message store partitions and Message Transfer Agent (MTA) queues reside on this shared file system.

N+1 Cluster

The N + 1 or “N over 1” model operates in a multi-node asymmetrical configuration. N logical host names and N shared disk arrays are required. A single backup node is reserved as a hot standby for all the other nodes. The backup node is capable of concurrently running Messaging Server from the N nodes.

The shared file system is also accessed by a failover node, shown in Figure 4-1 as NodeX. When any of the N nodes fail, The failover node, which had been standing by, becomes active, running the binaries of the failed node using the corresponding configuration and data in the shared file system. (The expectation is that only one node would fail at any given time.)

The key concept in upgrading Messaging Server in this cluster is to sequentially upgrade the N active nodes by failing over the node being upgraded to NodeX, while patching and re-configuring (running the `patch-config` script) Messaging Server for that node.

For example, consider the upgrade of Messaging Server on Node1. Messaging Server services on Node1 are failed over to NodeX while patching and re-configuration take place. In other words MS-X binaries use the MS-1 config and MS-1 data wile the MS-1 binaries are patched and re-configured. When this part of the upgrade is complete, the Messaging Server services are failed back over to Node1. At that point, the Messaging Server services are shut down briefly while the `install-newconfig` script is run, after which the services are restarted.

The procedure is repeated for each node in the cluster.

The steps for Node1, for example, are as follows.

1. Make sure the MS-X binaries are at the same patch level as Node1.
2. On NodeX, set up the MS-X binaries to use the MS-1 config/MS-1 data, using the `useconfig` utility.
3. On Node1, stop the MS-1 services being run by MS-1 binaries.
4. On NodeX, start up the MS-1 services using the MS-X binaries.

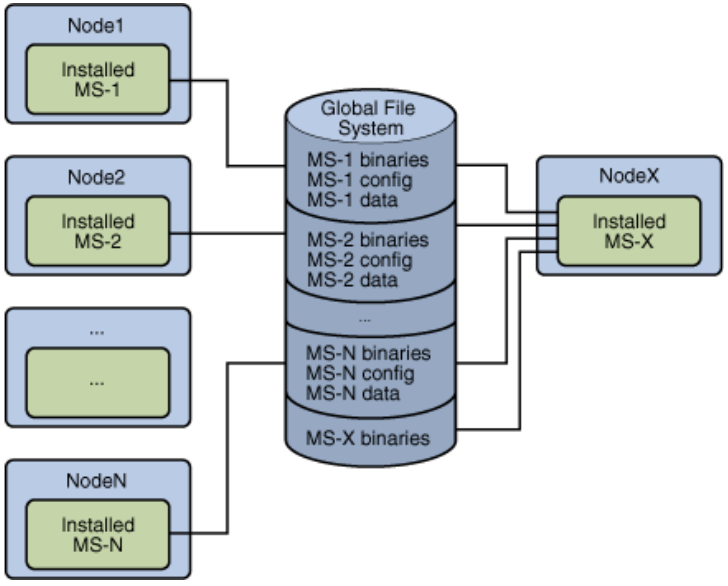
   (Step 2 on page 85 through Step 4 amount to a manual failover of MS-1 binaries to MS-X binaries.)
5. On Node 1, patch the MS-1 binaries and run the `patch-config` script.
6. On NodeX, stop the MS-1 services being run by MS-X binaries.

7. Run `install-newconfig` on MS-1.
8. Bring the MS-1 services back up.

Repeat the above steps for Node2 through NodeN of the cluster. Finally, apply the patch to MS-X on NodeX, so it is at the same patch level as all the other nodes.

Note that MS services are truly down only when `install-newconfig` is run. The other times that MS services are brought down are only for the brief period of a manual failover.

Figure 4-1 Messaging Server N+1 Cluster

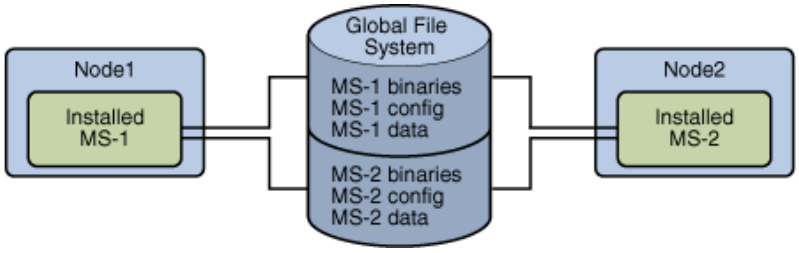


Symmetric HA

The basic symmetric or “dual services” high availability model consists of two hosting machines, each with its own logical IP address. Each logical node is associated with one physical node, and each physical node controls one disk array with two storage volumes. One volume is used for its local message store partitions and MTA queues, and the other is a mirror image of its partner’s message store partitions and MTA queues.

In the symmetric high availability mode (Figure 4-2), both nodes are active concurrently, and each node serves as a backup node for the other. Under normal conditions, each node runs only one instance of the messaging server.

Figure 4-2 Messaging Server Symmetric Cluster



In the case of symmetric high availability, The MS service being upgraded must be shut down for the duration of the patching and re-configuration. That is, MS-1 binaries need to be stopped during patching and cannot therefore be failed over to Node2.

Upgrading Messaging Server from Messaging Server 6.2 (Java Enterprise System 2005Q1)

The procedure for upgrading 2005Q1 Messaging Server 6.2 (Java Enterprise System 2005Q1) to 6.3 is the same as that for upgrading Messaging Server 6.2p3 (Java Enterprise System 2005Q4) to Messaging Server 6.3.

To upgrade Messaging Server 6.2 (Java Enterprise System 2005Q1) to Messaging Server 6.3, use the instructions in "Upgrading Messaging Server from Messaging Server 6.2p3 (Java Enterprise System 2005Q4)" on page 72, except substitute 6.2 (Java Enterprise System 2005Q1) wherever 6.2p3 (Java Enterprise System 2005Q4) is referenced.

Upgrading Messaging Server from Messaging Server 6.1 (Java Enterprise System 2004Q2)

The procedure for upgrading Messaging Server 6.1 (Java Enterprise System 2004Q2) to 6.3 is the same as that for upgrading Messaging Server 6.2p3 (Java Enterprise System 2005Q4) to Messaging Server 6.3, with a couple of exceptions, noted below.

Upgrade Messaging Server Dependencies

As compared to the upgrade from Messaging Server 6.2p3 (Java Enterprise System 2005Q4), the Messaging Server 6.1 (Java Enterprise System 2004Q2) to Messaging Server 6.3 pre-upgrade tasks should include the upgrading to 6.3 of all shared components (see Table 1-6 on page 31) and all locally-resident product components upon which Messaging Server depends:

1. **Shared Components.** Instructions for upgrading Communications Suite shared components to 6.3 are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Directory Server rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Access Manager (optional).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.
4. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in the Chapter 3, "Directory Preparation Tool" on page 53.

Messaging Server 6.1 (Java Enterprise System 2004Q2) Upgrade

The procedure for upgrading Messaging Server 6.1 (Java Enterprise System 2004Q2) to Messaging Server 6.3 depends on operating system platform.

Upgrading Messaging Server 6.1(Solaris)

To upgrade Messaging Server 6.1 (Java Enterprise System 2004Q2) to Messaging Server 6.3, use the instructions in "Upgrading Messaging Server 6.2p3 (Solaris)" on page 75, except substitute Messaging Server 6.1 (Java Enterprise System 2004Q2) wherever Messaging Server 6.2p3 (Java Enterprise System 2005Q4) is referenced.

Upgrading Messaging Server 6.1 (Linux)

The procedure documented below applies to all Messaging Server components that correspond to the same installed Messaging Server image on the computer where the upgrade is taking place.

CAUTION An upgrade from Messaging Server 6.1 Messaging Server 6.3 on Linux cannot be rolled back.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop all running Messaging Server components.

   msg-svr-base`/sbin/stop-msg`

3. If you have not already done so, synchronize shared components to 6.3.

   See "Upgrade Messaging Server Dependencies" on page 88.

4. Uninstall the 6.1 RPM packages.

   ```
   rpm -e --noscripts sun-messaging-lib-6.1-9 \
                      sun-messaging-store-6.1-9 \
                      sun-messaging-install-6.1-9 \
                      sun-messaging-core-6.1-9 \
                      sun-messaging-mmp-6.1-9 \
                      sun-messaging-sieveui-6.1-9 \
                      sun-messaging-webmail-6.1-9 \
                      sun-messaging-core-en-6.1-9 \
                      sun-messaging-mta-6.1-9
   ```

5. Install the RPMs for Messaging Server core and, if needed, localization patches in Table 4-5 on page 78.

   ```
   rpm -i sun-messaging-server-6.1-12.38.i386.rpm
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -q sun-messaging-server
   ```

   The version number of the newly installed RPM should be returned.

7. Save off your old 6.1 configuration.

   The configuration files are located at: *msg-svr-base*`/config`

8. Run the Messaging Server configuration program.

   ```
   cd msg-svr-base/sbin
   ./configure
   ```

9. Perform a manual merge of the 6.1 configuration values with the new 6.3 configuration entries.

10. Restart the Messaging Server components that were stopped in Step 2.

    msg-svr-base`/sbin/start-msg`

For further details, for example to change the HTTP port using the `configutil` command, see the Special Installation Instructions section of the patch 120230-17 `readme` file.

Verifying the Upgrade from Version 6.1

You can verify the upgrade from 6.1 to 6.3 in the same way as you can verify the upgrade from 6.2 to 6.3. See "Verifying the Messaging Server Upgrade to Version 6.3" on page 81.

Post-Upgrade Tasks for Upgrade from Version 6.1 to 6.3

The post-upgrade tasks in upgrading from 6.1 to 6.3 are the same as in upgrading from 6.2 to 6.3. See "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81.

Chapter 5

Upgrading Calendar Server

This chapter describes how to upgrade previous versions of Calendar Server to Sun Java System Calendar Server 6.3. The chapter provides an overview of upgrade considerations for the different upgrade paths. 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 *cal-svr-base*. At least part of this path might have been specified as an installation directory when Calendar Server was initially installed. If not, the installer assigned a default value.

The default value of *cal-svr-base* depends on operating system platform, as shown in the following table.

Path Name	Solaris OS	Linux OS
cal-svr-base	`/opt/SUNWics5`	`/opt/sun/calendar`

Overview of Calendar Server Upgrades

This section describes the following general aspects of Calendar Server that impact upgrading to Communications Suite 5:

About Calendar Server 6.3

This version of Calendar Server represents a minor release with respect to Calendar Server 6.2 (Java Enterprise System 2005Q4). It includes several new features such as support for attachments and LDAP groups, but no major functionality changes.

Calendar Server, however, has an private interface change that affects upgrade: the format for data in the Calendar Server store has changed to a new format.

Calendar Server Upgrade Roadmap

Table 5-1 shows the supported Calendar Server upgrade paths to Communications Suite 5. The table applies to both Solaris and Linux operating systems.

Table 5-1 Upgrade Paths to Communications Suite 5: Sun Java System Calendar Server 6.3

Release	Calendar Server Version	General Approach
Java Enterprise System 2005Q4	Sun Java System Calendar Server 6.2 2005Q4	Direct upgrade: Perform by applying patches.
Java Enterprise System 2005Q1	Sun Java System Calendar Server 6 2005Q1	Direct upgrade: Perform by applying patches.
Java Enterprise System 2004Q2	Sun Java System Calendar Server 6 2004Q2	Direct upgrade: Perform by applying patches.
Java Enterprise System 2003Q4	Sun ONE Calendar Server 6.0 (2003Q4)	Direct upgrade not certified: But can be performed by applying patches.
Pre-dates Java Enterprise System releases	All previous versions	No direct upgrade.

Calendar Server Data

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

Table 5-2 Calendar Server Data Usage

Type of Data	Location	Usage
Configuration data	`/etc/`*cal-svr-base*`/cal/config/ics.conf`	Configuration of Calendar Server
Dynamic application data	Calendar Server database: `/var/`*cal-svr-base*`/csdb`	Store calendar entries on behalf of users.
Directory schema	Directory Server user/group directory	For user attributes needed to support end users

Calendar Server Upgrade Strategy

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

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

Compatibility Issues

Calendar Server 6.3 does not introduce any changes in public interfaces, and is therefore backwardly compatible with Calendar Server 6.2 (Java Enterprise System 2005Q4), however the format of data in the Calendar Server Store has changed in this release, impacting the upgrade procedure.

Calendar Server Dependencies

Calendar Server dependencies on other Communications Suite components can impact the procedure for upgrading and re-configuring Calendar Server software. Changes in Calendar Server interfaces or functions, for example, could require upgraded version of components upon which Calendar Server depends. The need to upgrade such components depends upon the specific upgrade path.

Calendar Server has dependencies on the following Communications Suite components:

- **Shared components.** Calendar Server has dependencies on specific Communications Suite shared components (see Table 1-6 on page 31).
- **Directory Server.** Calendar Server accesses user data stored in Directory Server. As a result, Calendar Server upgrades might require extensions of directory schema.
- **Directory Preparation Tool.** Calendar Server uses the Directory Preparation Tool to prepare the directory to support Calendar Server functions.
- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Calendar Server can be configured to use Access Manager single sign-on capability.
- **Messaging Server (optional).** Calendar Server can be configured to use Messaging Server to provide messaging notifications of calendar events.
- **Delegated Administrator (optional).** Delegated Administrator is the preferred utility to use for provisioning users in Directory Server so that Calendar Server has access to the user data needed to provide calendar services.

Upgrading Calendar Server from Calendar Server 6.2 (Java Enterprise System 2005Q4)

This section includes information about upgrading Calendar Server from Calendar Server 6.2 (Java Enterprise System 2005Q4). The section covers the following topics:

Introduction

When upgrading to Calendar 6.3, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the 6.2 (Java Enterprise System 2005Q4) version and then using a migration utility to migrate the Calendar Server database to a new data format.
- **Upgrade Dependencies.** Calendar Server has dependencies on a number of Communications Suite shared components (see Table 1-6 on page 31); these are hard upgrade dependencies, therefore, these shared components are required to be upgraded.

In addition, Calendar Server is dependent upon Directory Server and optionally dependent on Access Manager, as described in "Calendar Server Dependencies" on page 93. These are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Calendar Server to 6.3.

However, Calendar Server has a hard upgrade dependency on Directory Preparation Tool; Directory Preparation Tool is required to prepare Directory Server for calendaring operations.

- **Backward Compatibility.** Calendar Server is backwardly compatible with Calendar Server 6.2 (Java Enterprise System 2005Q4) version, even though the format of Calendar Server data has changed.
- **Upgrade Rollback.** Rollback of the upgrade of Calendar Server to Calendar Server 6.2 (Java Enterprise System 2005Q4) is achieved by removing the patches applied during the upgrade and restoring the Calendar Server 6.2 (Java Enterprise System 2005Q4) database.
- **Platform Issues.** The general approach for upgrading Calendar Server is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Calendar Server 6.2 (Java Enterprise System 2005Q4) Upgrade

This section describes how to perform an upgrade of Calendar Server from Calendar Server 6.2 (Java Enterprise System 2005Q4) to Calendar Server 6.3 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

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

Verify Current Version Information

You can verify the current version of Calendar Server using the following command:

Solaris:

```
cd cal-svr-base/cal/bin
cscal version
```

Linux:

```
cd cal-svr-base/lib
cscal version
```

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

```
LD_LIBRARY_PATH= ./cscal -#
```

Table 5-3 Calendar Server Version Verification Outputs

Calendar Server Release	Calendar Server Version Number
2004Q2	`2004Q2`
2005Q1	`2005Q1`
2005Q4	`2005Q4`
As part of Communications Suite `5`	`6.3`

Upgrade Calendar Server Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 5. However, Calendar Server has a hard upgrade dependency only on Directory Preparation Tool. Upgrading of other Calendar Server 6.2 (Java Enterprise System 2005Q4) product components upon which Calendar Server depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Communications Suite shared components are provided in Chapter 2, “Upgrading Communications Suite Shared Components” on page 41.
2. **Directory Server.** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.

3. **Access Manager (optional).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.

4. **Directory Preparation Tool.** Directory Preparation Tool needs to have been run against Directory Server before configuring Calendar Server. If Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to 5 and use it to modify and extend the schema of Directory Server. See the Chapter 3, "Directory Preparation Tool" on page 53 for procedures.

5. **Messaging Server (optional).** Instructions for upgrading Messaging Server are provided in Chapter 4, "Upgrading Messaging Server" on page 67.

Back Up Calendar Server Data

The Calendar Server upgrade to 6.3 involves a change in Calendar Server data format requiring 2005Q4 data to be migrated to the 6.3 format. As a safety precaution, you should shut down Calendar Server and back up your Calendar Server store, located at:

`/var/`*`cal-svr-base`*`/csdb`

Obtain Required Configuration Information and Passwords

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

Upgrading Calendar Server 6.2 (2005Q4 - Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Calendar Server software to Communications Suite 5 takes into account the following considerations:

- Calendar Server should be shut down when patches are being applied to the installed image.
- In architectures in which different Calendar Server subcomponents reside on different computers, for example Calendar Server back-end store on one computer, and Calendar Server front-end processes (such as `cshttpd`) on another, the upgrade must be performed on all such computers.

- The Calendar Server upgrade applies to multiple subcomponents of Calendar Server on one computer using the same installed image.
- The Calendar Server upgrade patches for Solaris OS are shown in the following table:

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

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Calendar Server core	121657-17	121658-17
Calendar Server localization	117010-26	117011-26

1. Patch revision numbers are the minimum required for upgrade to Communications Suite 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 Calendar Server on the computer where the upgrade is taking place.

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

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Calendar Server if it is running.

   cal-svr-base`/cal/sbin/stop-cal`

4. If you have not already done so, upgrade shared component to Communications Suite 5.

   See "Upgrade Calendar Server Dependencies" on page 96.

5. Apply the appropriate Calendar Server core and, if needed, localization patches in Table 5-4, in that order.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrades were successful:

   ```
   showrev -p | grep ics
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Re-configure Calendar Server.

   ```
   cd cal-svr-base/sbin
   ./csconfigurator.sh -noconsole -nodisplay -novalidate
   ```

   The `-noconsole -nodisplay -novalidate` arguments pick up the existing Calendar Server 6.2 (Java Enterprise System 2005Q4) configuration values and perform the necessary re-configuration.

   If the Calendar Server 6.2 (Java Enterprise System 2005Q4) installation had been configured in non-hosted domain (legacy) mode, then the configurator gives the choice to either stay in that mode or switch to hosted domain mode, the default for Calendar Server 6.3. Switching to hosted domain mode is not reversible.

   The `csconfigurator` command is documented in the Calendar Server *Administration Guide,* http://docs.sun.com/doc/819-2433/6n4nlfjk2?a=view.

8. Move Calendar Server data files to a temporary location.

   ```
   mkdir /var/cal-svr-base/old_csdb
   mv /var/cal-svr-base/csdb/* /var/cal-svr-base/old_csdb
   ```

   where *old_csdb* is a temporary location.

9. Change permissions on the temporary location.

   ```
   chown -R icsuser:icsgroup /var/cal-svr-base/oldcsdb
   ```

10. Migrate the Calendar Server 6.2 (Java Enterprise System 2005Q4) data using the `csmigrate` migration tool.

    ```
    cd cal-svr-base/cal/sbin
    ./csmigrate -l max /var/cal-svr-base/old_csddb /var/cal-svr-base/csdb
    ```

    The general syntax for `csmigrate` is as follows:

    ```
    csmigrate [-q] [-d] [-l min|max] [-b backup_dir] source_dbdir target_dbdir
    ```

    Command options and operands are documented in the following table.

Table 5-5 `csmigrate` Command Options and Operands

Option/Operand	Description
`-q`	Specifies quiet mode, no print statements
`-d`	Specifies dry run mode, no new db written
`-l min|max`	Specifies log level. `csmigrate` creates the following log files: *cal-svr-base*`/logs/csmigrate.log` *cal-svr-base*`/logs/csmigrateError.log`
`-b` *backup_dir*	Specifies directory to which to back up the source database. The program works on te backed-up copy to prevent any damage to the source database. The default location is *source_dbdir*`/backup`.
source_dbdir	Directory where pre-migration database files are located
target_dbdir	Directory where post-migration files will be written

If you choose an arbitrary *target_dbdir* rather than `/var/`*cal-svr-base*`/csdb`, then you have to change the value of the `caldb.berkeleydb.homedir.path` property in the Calendar Server configuration file to point to that location.

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

```
LD_LIBRARY_PATH= ./csmigrate ...
```

11. Restart the Calendar Server that was stopped in Step 3.

cal-svr-base`/cal/sbin/start-cal`

Upgrading Calendar Server 6.2 (2005Q4 - Linux)

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

Upgrade Considerations (Linux)

The upgrade of Calendar Server software on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 97), except that the Linux upgrade patches differ from the Solaris patches.

The Calendar Server upgrade patches for Linux OS are shown in the following table:

Table 5-6 Patches[1] to Upgrade Calendar Server on Linux

Description	Patch ID and RPM names
Calendar Server core	121659-17 • `sun-calendar-core-6.3-11.9.i386.rpm` • `sun-calendar-api-6.3-11.9.i386.rpm`
Calendar Server localization	117852-26 • `sun-calendar-core-<locale>-6.3-1.7.i386.rpm`

1. Patch revision numbers are the minimum required for upgrade to Communications Suite 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 Calendar Server on the computer where the upgrade is taking place.

CAUTION An upgrade from Calendar Server 6.2 (Java Enterprise System 2005Q4) Calendar Server 6.3 5 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 5-6. Use this information to obtain the version numbers for the RPM.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Calendar Server if it is running.

   cal-svr-base`/sbin/stop-cal`

4. If you have not already done so, synchronize all shared component to Release 5.

   See "Upgrade Calendar Server Dependencies" on page 96.

5. Apply the core and, if needed, localization RPMs for Calendar Server in Table 5-6, in that order.

   ```
   rpm -Fvh sun-calendar-core-6.2-10.7.i386.rpm
   rpm -Fvh sun-calendar-api-6.2-10.7.i386.rpm
   rpm -Fvh sun-calendar-core-l10n-6.2-10.3.i386.rpm
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -qa | grep sun-calendar
   ```

   The new version numbers of the RPMs in Step 5 should be returned.

7. Re-configure Calendar Server.

   ```
   cd cal-svr-base/sbin
   ./csconfigurator.sh -noconsole -nodisplay -novalidate
   ```

   The `-noconsole -nodisplay -novalidate` arguments pick up the existing Calendar Server 6.2 (Java Enterprise System 2005Q4) configuration values and perform the necessary reconfiguration.

   If the Calendar Server 6.2 (Java Enterprise System 2005Q4) Calendar Server installation had been configured in non-hosted domain (legacy) mode, then the configurator gives the choice to either stay in that mode or switch to hosted domain mode, the default for 6.3. Switching to hosted domain mode is not reversible.

   The csconfigurator command is documented in the *Calendar Server Administration Guide,* `http://docs.sun.com/doc/819-2433/6n4nlfjk2?a=view`.

8. Move Calendar Server data files to a temporary location.

   ```
   mkdir /var/cal-svr-base/oldcsdb
   mv /var/cal-svr-base/csdb/* /var/cal-svr-base/oldcsdb
   ```

   where *oldcsdb* is a temporary location.

9. Change permissions on the temporary location.

   ```
   chown -R icsuser:icsgroup //var/cal-svr-base/oldcsdb
   ```

10. Migrate the Calendar Server 6.2 (Java Enterprise System 2005Q4) data using the `csmigrate` migration tool.

    ```
    cd cal-svr-base/sbin
    ./csmigrate -l max /var/cal-svr-base/old_csddb /var/cal-svr-base/csdb
    ```

    The general syntax for `csmigrate` is as follows:

    `csmigrate [-q] [-d] [-l min|max] [-b` *backup_dir*`]` *source_dbdir target_dbdir*

    Command options and operands are documented in the following table.

Table 5-7 `csmigrate` Command Options and Operands

Option/Operand	Description
`-q`	Specifies quiet mode, no print statements
`-d`	Specifies dry run mode, no new db written
`-l min|max`	Specifies log level. `csmigrate` creates the following log files: *cal-svr-base*`/logs/csmigrate.log` *cal-svr-base*`/logs/csmigrateError.log`
`-b` *backup_dir*	Specifies directory to which to back up the source database. The program works on te backed-up copy to prevent any damage to the source database. The default location is *source_dbdir*`/backup`.
source_dbdir	Directory where pre-migration database files are located
target_dbdir	Directory where post-migration files will be written

If you choose an arbitrary *target_dbdir* rather than `/var/`*cal-svr-base*`/csdb`, then you have to change the value of the `caldb.berkeleydb.homedir.path` property in the Calendar Server configuration file to point to that location.

11. Restart the Calendar Server that was stopped in Step 3.

    cal-svr-base`/sbin/start-cal`

Verifying the Upgrade

You can verify the upgrade of Calendar Server by checking the version numbers of patches as described in Step 6 on page 98 for Solaris platforms and Step 6 on page 102 for Linux.

You can also use the following commands:

Solaris:
cal-svr-base`/cal/sbin/cscal version`

Linux:
cal-svr-base`/sbin/cscal version`

See Table 5-3 on page 96 for version values.

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 98 and "Upgrade Procedure (Linux)" on page 101.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Calendar Server followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade to 6.3 of Calendar Server is pretty much the reverse of the procedure for upgrading to 6.3.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Calendar Server.

   cal-svr-base`/cal/sbin/stop-cal`

3. Remove the patches in Table 5-4 on page 98.

   `patchrm` *patch_ID*

4. Revert to the Calendar Server 6.2 (Java Enterprise System 2005Q4) database.

   Any new data added since the upgrade to 6.3 will be lost.

5. Restart Calendar Server.

   cal-svr-base`/cal/sbin/start-cal`

Multiple Instance Upgrades

In some deployment architectures Calendar Server is deployed on multiple computer systems to provide for high availability. For example, you have the Calendar Server Store component running in a Sun Cluster environment to provide high availability.

For Calendar Server instances running in a cluster environment, those instances share the same configuration. You need to apply Calendar Server upgrade patches to each of the instances, and for a Calendar Server 6.2 (Java Enterprise System 2005Q4) to 6.3 upgrade there is no re-configuration required.

Upgrading Calendar Server from Calendar Server 6 (Java Enterprise System 2005Q1)

The procedure for upgrading Calendar Server 6 (Java Enterprise System 2005Q1) to 6.3 is the same as that for upgrading Calendar Server 6.2 (Java Enterprise System 2005Q4) to 6.3.

To upgrade Calendar Server 6 (Java Enterprise System 2005Q1) to 6.3, use the instructions in "Upgrading Calendar Server from Calendar Server 6.2 (Java Enterprise System 2005Q4)" on page 94, except substitute Calendar Server 6 (Java Enterprise System 2005Q1) wherever 2005Q4 is referenced.

Upgrading from Calendar Server 6 (Java Enterprise System 2004Q2)

The procedure for upgrading Calendar Server 6 (Java Enterprise System 2004Q2) to 6.3 is very similar to that for upgrading Calendar Server 6.2 to 6.3, with the exception that the pre-upgrade tasks must include the synchronizing of all shared components (see Table 1-6 on page 31) to 6.3 and upgrading all locally-resident product components upon which Calendar Server depends:

1. **Shared Components.** Instructions for updating Communications Suite shared components to **5** are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Directory Server rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Access Manager (optional).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.
4. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Messaging Server, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in the Chapter 3, "Directory Preparation Tool."

To upgrade Calendar Server 6 (Java Enterprise System 2004Q2 Calendar Server to 6.3, use the instructions in "Upgrading Calendar Server from Calendar Server 6.2 (Java Enterprise System 2005Q4)" on page 94, except substitute 2004Q2 wherever Calendar Server 6.2 (Java Enterprise System 2005Q4) is referenced.

In addition, Calendar Server 6.2 (Java Enterprise System 2005Q4) to Calendar Server 6.3 upgrade requires the post-upgrade task of configuring Calendar Server hot backup, accomplished by adding hotbackup parameters to the Calendar Server `ics.conf` configuration file. The instructions for this post-upgrade re-configuration can be found at the following location: http://docs.sun.com/doc/819-2433/6n4nlfjnq?a=view

Chapter 6

Upgrading Communications Express

This chapter describes how to upgrade previous versions of Communications Express to Sun Java System Communications Express 6.3. The chapter provides an overview of upgrade considerations for the different upgrade paths. 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 *CommsExpress-base* and *CommsExpressDeploy-base*. At least part of this path might have been specified as an installation directory when Communications Express was initially installed. If not, the Communications Suite installer assigned a default value.

The default value of *CommsExpress-base* depends on operating system platform, as shown in the following table.

Path Name	Solaris OS	Linux OS
CommsExpress-base	`/opt/SUNWuwc`	`/opt/sun/uwc`
CommsExpressDeploy-base	`/var/opt/SUNWuwc`	`/var/opt/sun/uwc`

Overview of Communications Express Upgrades

This section describes the following general aspects of Communications Express that impact upgrade:

About Communications Express 6.3

Communications Express 6.3 represents a major release with respect to Communications Express 6.2 (Java Enterprise System 2005Q4). Communications Express supports features such as password encryption in configuration files, configuration tool automation, shared and multiple address books, and calendar attachments.

In addition, several of its internal implementations have been changed, impacting the upgrade procedure:

- The use of a mail proxy servlet filter to de-couple Communications Express from the Messaging Server MEM component. The MEM component has changed this release, and need not run on the same computer as Communications Express. As a result of this de-coupling, you need to migrate customizations from the current MEM location to a new Communications Express location. (For information about changes in Messaging Server MEM, see "About Messaging Server 6.3" on page 68.)

NOTE You can upgrade Messaging Server without upgrading Communications Express. However, if you upgrade both Messaging Server and Communications Express, you must upgrade the back-end servers (Message Store) before upgrading the front-end (Communications Express clients). Upgrade the Webmail Server only when you upgrade the Communications Express clients; do not upgrade the Webmail Server before upgrading the Communications Express clients or your upgrade will be unsuccessful.

- The previous dependency of Communications Express upon Access Manager (Access Manager SDK) to support Schema 2 directory lookups has been removed in version 6.3. However Access Manager (Access Manager SDK) is still needed to provide single sign-on authentication and authorization services for end users.

Changes in Communications Express File Locations

Prior to Communications Suite 5, the `html/js/gif/jpg` files used by Communications Express to display the Mail User Interface resided in the Messaging Server packaging (in the *msg-svr-base*`/config/html` directory). Beginning with Communications Suite 5, Communications Express picks up these files from the Communications Express packaging. If you are upgrading to Communications Suite 5, you will use a Communications Express migration script to move the user customizations from the Messaging Server area to the Communications Express area. This is a one-time migration move.

For a Messaging Server upgrade from a pre-Communications Suite 5 version to a Communications Suite 5 version (or later), the Communications Express files in Messaging Server packaging will be removed. A copy of the entire `config/html` area is placed in *msg-svr-base*`/data/preJES5/config/html` which the Communications Express migration script will use. In addition, a copy of the previous version of saved files in `lib/config-templates/html` will be saved to *msg-svr-base*`/data/preJES5/lib/config-templates/html.`

When upgrading Communications Express and its Messaging front-end server from pre-Communications Suite 5 to Communications Suite 5 (or later), it is recommended that you upgrade Messaging Server first.

Note that in Communications Suite 5, you no longer need to have the `mshttpd` process running on the back-end servers since `mshttpd` can talk IMAP to the back-end servers. In addition, in Communications Express no longer requires that the Messaging Server `mshttpd` be running on the same machine as Communications Express since the `html/js/gif` files are now being loaded from the Communications Express area.

Communications Express Upgrade Roadmap

Table 6-1 shows the supported Communications Express upgrade paths to Communications Express 6.3. The table applies to both Solaris and Linux operating systems.

Table 6-1 Upgrade Paths to Communications Express 6.3

Release	Communications Express Version	General Approach	Re-configuration Required
Java Enterprise System 2005Q4	Sun Java System Communications Express 6.2 2005Q4	Direct upgrade: Perform by applying patches and re-configuring webmail component.	Configuration files updated and customizations migrated.
Java Enterprise System 2005Q1	Sun Java System Communications Express 6 2005Q1	Direct upgrade: Perform by applying patches and re-configuring webmail component.	Configuration files updated and customizations migrated.
Java Enterprise System 2004Q2	Sun Java System Communications Express 6 2004Q2	Direct upgrade: Perform by applying patches and re-configuring webmail component.	Configuration files updated and customizations migrated.
Java Enterprise System 2003Q4	None	No upgrade.	
Pre-dates Java Enterprise System releases	None	No upgrade.	

Communications Express Data

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

Table 6-2 Communications Express Data Usage

Type of Data	Location	Usage

Configuration data:	Local configuration directory
	`var/`*CommsExpress-base*`/WEB-INF/config/uwcauth.properties`
	`var/`*CommsExpress-base*`/WEB-INF/config/uwcconfig.properties`
	`var/`*CommsExpress-base*`/WEB-INF/config/uwclogging.properties`
	msg-svr-base`/config/msg.conf`
	and other configuration files for configuring Messaging Server MEM (webmail) have been removed from version 6.3 because of a different webmail implementation
		Configuration of Communications Express, including webmail

Web container access control and configuration files	Web Server 7.0 (Java Enterprise System 5) `server.policy` and `server.xml` files in *WebServer7Config-base*`/https-`*configName*`/config` Web Server 6.*x* (JES 2005Q1, 2005Q4, and Communications Suite 5) `server.policy` and `server.xml` files in *WebServer6-base*`/https-`*hostname*`/config` Application Server 8.*x* (JES 2005Q1, 2005Q4, and Communications Suite 5): `server.policy` and `domain.xml` files in *AppServer8Config-base*`/domains/`*domainName*`/config` Application Server 7.*x* (JES 2004Q2): `server.policy` and `server.xml` files in *AppServer7Config-base*`/domains/`*domainName*`/config`	Configuration of Communications Express web container instance.
Directory schema	Directory Server user/group directory	For user attributes needed to support end users

Communications Express Upgrade Strategy

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

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

Compatibility Issues

Communications Express does not introduce any public interface changes and is therefore backwardly compatible with Communications Express 6.2 (Java Enterprise System 2005Q4), however changes in internal implementations impact the upgrade procedure.

In particular, the de-coupling of Communications Express from the Messaging Server MEM webmail server component requires you to migrate customizations to a new location, and allows you to move the (webmail server) MEM component to a remote computer if you wish.

Communications Express Dependencies

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

Communications Express has dependencies on the following Communications Suite components:

- **Shared components.** Communications Express has dependencies on specific Communications Suite shared components (see Table 1-6 on page 31).
- **Web Container.** Communications Express depends upon web container services, which can be provided either by Java Enterprise System Web Server or Java Enterprise System Application Server.
- **Access Manager (or Access Manager SDK).** Access Manager (Access Manager SDK) provide for single sign-on authentication and authorization services for end users. This is an optional dependency.
- **Messaging Server.** Communications Express is used to provide web-based access to Messaging Server. In fact Communications Express directly employs the Messaging Server MEM component to access other Messaging Server back-end components, such as the Messaging Server Store and MTA components.

- **Calendar Server.** Communications Express is used to provide web-based access to Calendar Server.
- **Directory Server.** Communications Express stores configuration data and also accesses user data stored in Directory Server. As a result, Communications Express upgrades might require upgrades of Directory Server or extensions of directory schema.
- **Directory Preparation Tool.** Communications Express uses the Directory Preparation Tool to prepare Directory Server to support Communications Express functions. As a result, Communications Express upgrades might depend upon preparation of the directory to support new functions.

Web Container Upgrade Scenarios

Communications Express can be deployed in a web container provided by either Web Server or Application Server. As a result, the upgrade of Communications Express can be complicated by the possibility of also having upgraded 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 6-3 Web Container Upgrade Scenarios for Communications Express Upgrade

Scenario	Web Container in which Communications Express is Originally Deployed	Web Container in which Communications Express is Deployed After Upgrade	Applicable Communications Express Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 6.*x*	2005Q1 2005Q4
Scenario 2	Web Server 6.*x*	Web Server 7.0	2004Q2 2005Q1 2005Q4
Scenario 3	Application Server 8.1	Application Server 8.1	2005Q1 2005Q4
Scenario 4	Application Server 8.1	Application Server 8.2	2005Q1 2005Q4
Scenario 5	Application Server 7*x*	Application Server 8.2	Not Supported

You must be careful when upgrading Communications Express (for example when re-configuring or re-deploying Communications Express) to provide values appropriate to the upgrade scenario in Table 6-3 that applies, especially when there is a major version upgrade of the web container.

Upgrading from Communications Express 6.2 (Java Enterprise System 2005Q4)

This section includes information about upgrading from Communications Express 6.2 (Java Enterprise System 2005Q4). The section covers the following topics:

Introduction

When upgrading Communications Express 6.2 (Java Enterprise System 2005Q4) Communications Express 6.3, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the Communications Express 6.2 (Java Enterprise System 2005Q4) version. Re-configuration of Communications Express, including the new webmail implementation, is achieved using configuration utilities.
- **Upgrade Dependencies.** While Communications Express has dependencies on a number of Communications Suite shared components (see Table 1-6 on page 31), Calendar Server is compatible with the Communications Express 6.2 (Java Enterprise System 2005Q4) versions of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Calendar Server to 6.3.

  However, Communications Express has hard upgrade dependencies on both Calendar Server and Messaging Server, for which it provides web-based access. Both Calendar Server and Messaging Server must be upgraded to version 6.3 before Communications Express can be upgraded to 6.3.

  In addition, Communications Express 6.3 has a hard upgrade dependency on Directory Preparation Tool; Directory Preparation Tool is required to prepare Directory Server for Communications Express functions.

Communications Express 6.3 is also dependent upon a web container, as described in "Communications Express Dependencies" on page 112. However, this is a soft upgrade dependency; upgrade of the web container is optional with respect to upgrade of Communications Express to 6.3.

- **Backward Compatibility.** Communications Express 6.3 is backwardly compatible with Communications Express 6.2 (Java Enterprise System 2005Q4), even though there have been changes in its internal implementation.
- **Upgrade Rollback.** Rollback of the 6.3 upgrade of Communications Express to Communications Express 6.2 (Java Enterprise System 2005Q4) is achieved by removing changes to local configuration files and then removing the patches applied during the upgrade.
- **Platform Issues.** The general approach for upgrading Communications Express is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

Communications Express 6.2 (Java Enterprise System 2005Q4) Upgrade

This section describes how to perform an upgrade from Communications Express 6.2 (Java Enterprise System 2005Q4) on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Communications Express you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Communications Express by accessing the Communications Express login page, which shows the current version number.

```
http://hostName:port/uwc/auth
```

Table 6-4 Communications Express Version Verification Outputs

Communications Express Release	Communications Express Version Number
2004Q2	`Sun Java System Communications Express 6 2004Q2`
2005Q1	`Sun Java System Communications Express 6 2004Q2 (This version string is in error; it should be 2005Q1)`
2005Q4	`Sun Java System Communications Express 6 2005Q4`
As part of Communications Suite 5	`Sun Java System Communications Express 6.3`

Upgrade Communications Express Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 6.3. However, Communications Express has hard upgrade dependencies only on Messaging Server, Calendar Server, and Directory Preparation Tool. Upgrading of other Communications Suite 5 product components upon which Communications Express depends is therefore optional.

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

1. **Shared Components.** Instructions for upgrading Communications Suite shared components are provided in Chapter 2, “Upgrading Communications Suite Shared Components” on page 41.
2. **Directory Server.** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in the *Java Enterprise System Upgrade Guide*.

4. **Directory Preparation Tool.** Directory Preparation Tool needs to have been run against Directory Server before configuring Communications Express. If the Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool and use it to modify and extend the schema of Directory Server. See the Chapter 3, “Directory Preparation Tool” on page 53 for procedures.

5. **Messaging Server.** Messaging Server components need to be upgraded to 6.3 to support Communications Express 6.3. Instructions for upgrading Messaging Server to 6.3 are provided in Chapter 4, “Upgrading Messaging Server” on page 67.

6. **Calendar Server.** Calendar Server components need to be upgraded to support Communications Express. Instructions for upgrading Calendar Server to 6.3 are provided in Chapter 5, “Upgrading Calendar Server” on page 91.

Back Up Communications Express Data

The Communications Express upgrade from Communications Express 6.2 (Java Enterprise System 2005Q4) to 6.3 requires re-configuration of the webmail implementation. The local changes can be rolled back, but as a precaution you can back up the `uwcconfig.properties` and `uwcauth.properties` files.

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. Depending on the web container you are using, you might need the corresponding administrator user ID and password.

Upgrading Communications Express 6.2 (Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Communications Express software takes into account the following considerations:

- Communications Express includes components used to provide web-based access to Calendar Server and Messaging Server back-end components. The upgrade patches encompass changes in all these components.
- The Messaging Server MEM component used to implement webmail has been de-coupled from Communications Express. As a result, the (webmail server) MEM need not run locally. (To move the MEM component to a remote computer, see “Post-Upgrade Tasks” on page 128.) In addition, customizations need to be moved from their current MEM location to a new Communications Express location.

- All Communications Express components should be deployed to the same web container. The web container should be shut down before patches are applied to the installed image.
- The Communications Express upgrade patches for Solaris OS are shown in the following table:

Table 6-5 Patches[1] to Upgrade Communications Express on Solaris

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Communications Express core	122793-12	122794-12
Communications Express localization	118042-19	118042-18

1. Patch revision numbers are the minimum required for upgrading Communications Express. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Solaris)

The procedure documented below applies to all Communications Express components on the computer being updated.

1. Obtain the required patches, based on Table 6-5.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Communications Express by stopping its web container.

   Web Server 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. If you have not already done so, synchronize all shared comonents to Release 5.

   See "Upgrade Communications Express Dependencies" on page 116.

5. Apply the appropriate Communications Express core and, if needed, localization patches in Table 6-5, in that order.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrades were successful:

   ```
   showrev -p | grep uwc
   ```

   The output should return the versions of patch IDs applied in Step 5.

7. Migrate configuration data from existing configuration files to the 6.3 configuration files.

   a. Move configuration data from existing configuration files to 6.3 configuration files for both core and, if needed, localization patch files.

   ```
   cd CommsExpress-base/sbin
   ./patch-config CommsExpress-base/install/patch/patch_ID
                  [-m MessagingServer-base]
   ```

   where the `-m` option migrates customizations from their current MEM location to a new location. If you omit this option, you will be prompted for this location the first time you run `patch-config`, but not on subsequent runs.

   The `patch-config` script prompts you for the current Communications Express configuration directory and then backs up the existing configuration files. It then prompts for new configuration parameters needed for the upgrade:

Table 6-6 Information Requested by `patch-config` Script

Information	Description
Webmail proxy admin id	Admin user id for webmail proxy authentication: must match setting on Messaging Server Store (see "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81)
Webmail proxy admin password	Admin password id for webmail proxy authentication: must match setting on Messaging Server Store (see "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81)
Webmail authentication only ssl (true/false)	Enable proxy authentication to webmail from CE over SSL
Webmail port [80]	Webmail server non-ssl port
Webmail ssl port (Optional) [443]	Webmail server ssl port (needed only if webmail authentication is set to `true`)
Directory Server Base DN	Required for direct access to Directory Server (rather than by way of Access Manager SDK)
Access Manager SSL login URL	Support SSO using AM SSL login URL

The `patch-config` script then merges configuration parameter values in these files with template configuration files to create new candidate configuration files. You should check these new files for possible conflicts, as described in the Special Installation Instructions section of the patch `readme` file.

b. Install the candidate configuration files, making them the active configuration. Do this for both core and, if needed, localization files

```
./install-newconfig CommsExpress-base/install/patch/patch_ID
```

This command installs the new configuration files in their correct locations.

8. Reconfigure the Messaging Server MEM to enable integration with Communications Express.

The `configutil` command below is needed in addition to those used when upgrading Messaging Server MEM to version 6.3, as documented in Chapter 4, "Upgrading Messaging Server," in the section "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81.

```
MessagingServer-base/sbin/configutil
-o local.webmail.sso.uwcenabled -v 1
```

The `configutil` commands below are also available if required:

```
MessagingServer-base/sbin/configutil
-o service.http.allowadminproxy -v yes
```

```
MessagingServer-base/sbin/configutil
-o local.service.http.proxy.admin -v  admin
```

```
MessagingServer-base/sbin/configutil
-o local.service.http.proxy.adminpass -v orionorion
```

```
MessagingServer-base/sbin/configutil
-o local.webmail.sso.uwcenabled -v 1
```

9. Remove the JSP class cache for Communications Express that is maintained by the web container.

For the procedure, see the documentation for your web container (Web Server or Application Server).

10. Restart Communications Express by restarting its web container.

Web Server 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:*
To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

AppServer8-base/sbin/asadmin start-domain --user *admin_ID*
--password *password domainName*

AppServer8-base/sbin/asadmin start-node-agent
--startinstances=false --user *admin_ID* --password *password*
nodeagentName

AppServer8-base/sbin/asadmin start-instance --user *admin_ID*
--password *password instanceName*

In the above commands, the following conventions are used:

- *nodeAgentName* has the form *hostName*
- The default *domainName* is domain1
- The default *instanceName* is server1

Upgrading Communications Express 6.2 (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Communications Express software on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 117), except that the Linux upgrade patches differ from the Solaris patches.

The Communications Express upgrade patches for Linux OS are shown in the following table:

Table 6-7 Patches[1] to Upgrade Communications Express on Linux

Description	Patch ID and RPM names
Communications Express core	122795-12 • `sun-uwc-6.3-10.5.i386.rpm`
Communications Express localization	118044-19 • `sun-uwc-<locale>-6.3-1.9.i386.rpm`

1. Patch revision numbers are the minimum required for Communications Express upgrade. If newer revisions become available, use the newer ones instead of those shown in the table.

Upgrade Procedure (Linux)

The procedure documented below applies to all Communications Express components on the computer being updated.

CAUTION An upgrade from Communications Express 6.2 (Java Enterprise System 2005Q4) to Communications Express 6.3 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 6-7. Use this information to obtain the version numbers for the RPM.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Communications Express 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. If you have not already done so, synchronize all shared components to Release 5.

   See "Upgrade Communications Express Dependencies" on page 116.

5. Apply the RPMs corresponding to the core and, if needed, localization patches for Communications Express in Table 6-7, in that order.

   For example:

   ```
   rpm -Fvh sun-uwc-6.1-10.2.i386.rpm
   ```

   To apply localization patches, additional `prepatch` and `postpatch` scripts need to be executed. For example:

   ```
   sh 118-044-19.prepatch
   ```

   ```
   rpm -Fvh sun-uwc-<locale>-6.1-11.11.i386.rpm
   ```

   ```
   sh 118044-19.postpatch
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -qa | grep uwc
   ```

   The output should return the version of RPM in Step 5.

7. Migrate configuration data from existing configuration files to 6.3 configuration files.

   a. Move configuration data from existing configuration files to 6.3 candidate configuration files for both core and, if needed, localization patch files.

   ```
   cd CommsExpress-base/sbin
   ./patch-config CommsExpress-base/install/patch/patch_ID
                  [-m MessagingServer-base]
   ```

   where the `-m` option migrates customizations from their current MEM location to a new location. If you omit this option, you will be prompted for this location the first time you run `patch-config`, but not on subsequent runs.

   The `patch-config` script prompts you for the current configuration directory and then backs up the existing configuration files. It then prompts for new configuration parameters needed for the upgrade:

Table 6-8 Information Requested by `patch-config` Script

Information	Description
Webmail proxy admin id	Admin user id for webmail proxy authentication: must match setting on Messaging Server Store (see "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81)
Webmail proxy admin password	Admin password id for webmail proxy authentication: must match setting on Messaging Server Store (see "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81)
Webmail authentication only ssl (true/false)	Enable proxy authentication to webmail from CE over SSL
Webmail port [80]	Webmail server non-ssl port
Webmail ssl port (Optional) [443]	Webmail server ssl port (needed only if webmail authentication is set to `true`)
Directory Server Base DN	Required for direct access to Directory Server (rather than by way of Access Manager SDK)
Access Manager SSL login URL	Support SSO using AM SSL login URL

The `patch-config` script then merges configuration parameter values in these files with the 6.3 template configuration files to create new configuration files. You should check these new files for possible conflicts, as described in the Special Installation Instructions section of the patch `readme` file.

b. Install the 6.3 candidate configuration files, making them the active configuration. Do this for both core and, if needed, localization files

```
./install-newconfig CommsExpress-base/install/patch/patch_ID
```

This command installs the new 6.3 configuration files in their correct 6.3 locations.

8. Reconfigure the Messaging Server MEM to enable integration with Communications Express.

The `configutil` command below is needed in addition to those used when upgrading Messaging Server to 6.3, as documented in Chapter 4, "Upgrading Messaging Server," in the section "Post-Upgrade Tasks for Messaging Server Version 6.3" on page 81.

```
msg-svr-base/sbin/configutil
-o local.webmail.sso.uwcenabled -v 1
```

9. Remove the JSP class cache for Communications Express that is maintained by the web container.

For the procedure, see the documentation for your web container (Web Server or Application Server).

10. Restart Communications Express by restarting its web container.

Web Server 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:*
To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

AppServer8-base/sbin/asadmin start-domain --user *admin_ID*
--password *password domainName*

AppServer8-base/sbin/asadmin start-node-agent
--startinstances=false --user *admin_ID* --password *password*
nodeagentName

AppServer8-base/sbin/asadmin start-instance --user *admin_ID*
--password *password instanceName*

In the above commands, the following conventions are used:

- *nodeAgentName* has the form *hostName*
- The default *domainName* is domain1
- The default *instanceName* is server1

Verifying the Upgrade

You can verify the current version of Communications Express by accessing the Communications Express login page, which shows the current version number.

`http://`*hostName*`:`*port*`/uwc/auth`

See Table 6-4 on page 116 for version values.

Once logged in, check the upgraded user interface for the new mail tab and old email and calendar events (if you are using those channels).

Also, you can check the log files for the various steps in the upgrade procedure:

CommsExpress-base`/install/patch/`122795-12
CommsExpress-base/install/patch/122795-12/`patch-config_20050729164754.log`
CommsExpress-base/install/patch/122795-12/`install-newconfig_20050729164838.log`

Post-Upgrade Tasks

As mentioned in "About Communications Express 6.3" on page 108, the (webmail server) MEM component has been de-coupled from Communications Express so that the MEM can reside remotely.

If you wish to move the MEM component to a remote computer, you need to specify the hostname and port number of the remote (webmail server) MEM component by manually editing the following configuration file:

`var/`*CommsExpress-base*`/WEB-INF/config/uwcconfig.properties`

The relevant parameters are shown in the table below.

Table 6-9 Webmail Configuration Parameters in uwcconfig.properties File

Parameter	Description
mail.deployed (true/false)	Set to `true` if mail is deployed in Communications Express. Default: `true`
webmail.cookiename	If `local.service.http.cookiename` is set in webmail then value of this parameter should have the same value. Otherwise set the value of this parameter to `webmailsid`.
webmail.host	mail HTTP Server hostname
webmail.port	mail HTTP Server port number. Default: `81`
webmail.securedproxyauth (true/false)	Set to `true` if proxy authentication will be done over ssl Set to `false` if proxy authentication will be non-ssl
webmail.proxyadmin	Proxy Admin user id

Table 6-9 Webmail Configuration Parameters in uwcconfig.properties File *(Continued)*

Parameter	Description
webmail.proxyadmin	Proxy Admin user password
webmail.ssl.port	mail HTTPS (SSL) server port number. Default: 443

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Communications Express followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade from 6.3 of Communications Express is pretty much the reverse of the procedure for upgrading to 6.3. The re-configurations are rolled back and the patches are removed.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Communications Express by stopping its web container.

   Web Server 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*

3. Roll back the re-configuration performed in Step 7 on page 120.

   The approach depends on that used to perform the re-configuration:

   Remote Messaging Server MEM.

   a. Start up the Communications Express configurator script.

      CommsExpress-base/sbin/uwc-config

   b. Revert configuration information to Communications Express 6.2 (Java Enterprise System 2005Q4) values.

   Local Messaging Server MEM.

   cd *CommsExpress-base*/sbin
   ./uninstall-newconfig *CommsExpress-base*/install/patch/*patch_ID*

4. Remove the patches in Table 6-5 on page 118.

   patchrm *patch_ID*

5. Restart Communications Express by restarting its web container.

   Web Server:
   WebServer-base/https-*instanceName*/start

   Application Server *8.x:*
   To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

   AppServer8-base/sbin/asadmin start-domain --user *admin_ID*
   --password *password* *domainName*

   AppServer8-base/sbin/asadmin start-node-agent
   --startinstances=false --user *admin_ID* --password *password*
   nodeagentName

   AppServer8-base/sbin/asadmin start-instance --user *admin_ID*
   --password *password* *instanceName*

   In the above commands, the following conventions are used:

   - *nodeAgentName* has the form *hostName*
   - The default *domainName* is domain1
   - The default *instanceName* is server1

Multiple Instance Upgrades

In some deployment architectures Communications Express is deployed on multiple computer systems to provide for scalability and to improve availability. For example, you might have Communications Express components running on multiple computers with a load balancer to distribute the load.

In the case of load-balanced instances of Communications Express, you can perform a rolling upgrade in which you upgrade the Communications Express instances sequentially without interrupting service. You upgrade each instance of Communications Express while the others remain running. You perform the upgrade of each instance as described in "Communications Express 6.2 (Java Enterprise System 2005Q4) Upgrade" on page 115.

Upgrading from Communications Express 6 (Java Enterprise System 2005Q1)

The procedure for upgrading Communications Express 6 (Java Enterprise System 2005Q1) to Communications Express 6.3 is the same as that for upgrading Communications Express 6.2 (Java Enterprise System 2005Q4) Communications Express to 6.3.

To upgrade Communications Express 6 (Java Enterprise System 2005Q1) to Communications Express 6.3, use the instructions in "Upgrading from Communications Express 6.2 (Java Enterprise System 2005Q4)" on page 114, except substitute Communications Express 6 (Java Enterprise System 2005Q1) wherever Communications Express 6.2 (Java Enterprise System 2005Q4) is referenced.

Upgrading from Communications Express 6 (Java Enterprise System 2004Q2)

The procedure for upgrading Communications Express 6 (Java Enterprise System 2004Q2) to Communications Express 6.3 is the same as that for upgrading Communications Express 6.2 (Java Enterprise System 2005Q4) to Communications Express 6.3, with a couple of exceptions, noted below.

NOTE This section applies to the case in which Communications Express is deployed in a 2004Q2 Web Server web container, but does not apply to the case in which Communications Express is deployed in a 2004Q2 Application Server web container. The latter case is not currently supported.

Upgrade Communications Express Dependencies

The pre-upgrade tasks for upgrading Communications Express 6 (Java Enterprise System 2004Q2) to Communications Express 6.3 are similar to those for upgrading Communications Express 6.2 (Java Enterprise System 2005Q4) to Communications Express 6.3, with the exception that the upgrade of Communications Express dependencies must include the upgrading to 6.3 of all shared components (see Table 1-6 on page 31) and all locally-resident product components upon which Communications Express depends.

When upgrading Communications Express dependencies, they should be upgraded in the following order, all before you upgrade Communications Express. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for upgrading Communications Suite shared components are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Directory Server rarely resides on the same computer as Communications Express, however, instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Web Container Software.** Instructions for upgrading Web Server are provided in the *Java Enterprise System Upgrade Guide*.
4. **Access Manager (or Access Manager SDK).** The previous dependency of Communications Express upon Access Manager (Access Manager SDK) to provide authentication and authorization services for end users, including single sign-on, has been removed in 6.3.
5. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Communications Express, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in the Chapter 3, "Directory Preparation Tool."

6. **Messaging Server.** Messaging Server MTA needs to be upgraded to 6.3 to support Communications Express 6.3. Instructions for upgrading Messaging Server to 6.3 are provided in "Upgrading Messaging Server from Messaging Server 6.1 (Java Enterprise System 2004Q2)" on page 87

7. **Calendar Server.** Calendar Server rarely resides on the same computer as Communications Express, however, instructions for upgrading Calendar Server to 6.3 are provided in "Upgrading from Calendar Server 6 (Java Enterprise System 2004Q2)" on page 105

Communications Express 6 (Java Enterprise System 2004Q2) Upgrade

The procedure for upgrading from Communications Express 6 (Java Enterprise System 2004Q2) to 6.3 depends on the web container in which you are deploying Communications Express software.

Upgrading Communications Express 6 (Java Enterprise System 2004Q2): Web Server Web Container

To upgrade Communications Express 6 (Java Enterprise System 2004Q2) to Communications Express 6.3, when deploying into a Web Server web container that has been upgraded to 6.3, follow the instructions in "Upgrading Communications Express 6.2 (Solaris)" on page 117 or "Upgrading Communications Express 6.2 (Linux)" on page 122, except substitute Communications Express 6 (Java Enterprise System 2004Q2) wherever Communications Express 6.2 (Java Enterprise System 2005Q4) is referenced.

Upgrading Communications Express 6 (Java Enterprise System 2004Q2): Application Server Web Container

The upgrade of Communications Express 6 (Java Enterprise System 2004Q2) to Communications Express 6.3, when deploying into an Application Server web container that has been upgraded to 6.3, is not currently supported.

Chapter 7

Upgrading Instant Messaging

This chapter describes how to upgrade Instant Messaging to Communications Suite 5: Sun Java System Instant Messaging 7.2. The chapter provides an overview of upgrade considerations for the different upgrade paths supported by Communications Suite 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 *InstantMessaging-base*. At least part of this path might have been specified as an installation directory when Instant Messaging was initially installed. If not, the installer assigned a default value.

The default value of *InstantMessaging-base* depends on operating system platform, as shown in the following table.

Path Name	Solaris OS	Linux OS
InstantMessaging-base	`/opt/SUNWiim`	`/opt/sun/im`
InstantMessagingRuntime-base	`/var/opt/SUNWiim`	`/var/opt/sun/im`

Overview of Instant Messaging Upgrades

This section describes the following general aspects of Instant Messaging that impact upgrade:

About Communications Suite 5 Instant Messaging

Communications Suite 5 Instant Messaging represents a minor release with respect to 2005Q4. The major new functionality in Communications Suite 5 is support for HTML-based clients, in addition to the Java-based Messenger client. This support requires a new `httpbind` gateway feature that requires post-upgrade configuration to enable. The gateway was introduced in the 2006Q1 IFR release.

In addition, Instant Messaging has a private interface change that affects upgrade: the Multiplexor protocol used between the Instant Messaging Multiplexor and server has changed. This version also includes some new administrative interfaces.

Communications Suite 5 Instant Messaging represents only a bug-fix update with respect to the Interim Feature Release (2006Q1) that followed 2005Q4.

NOTE Instant messaging has backward compatibility problems with earlier clients/API-based applications. It is preferable that the clients are linked with the newer version of the API that is bundled with the release. (Custom clients could move to `org.netbeans.lib.collab` API from `com.sun.im.service`.)

Instant Messaging Upgrade Roadmap

Table 7-1 shows the supported Instant Messaging upgrade paths. The table applies to both Solaris and Linux operating systems.

Table 7-1 Upgrade Paths to Communications Suite 5: Sun Java System Instant Messaging 7.2

Release	Instant Messaging Version	General Approach	Re-configuration Required
Java Enterprise System 2006Q1	Sun Java System Instant Messaging 7 2006Q1	Direct upgrade: Performed by applying patches.	None except for enabling new redirect server if desired.
Java Enterprise System 2005Q4	Sun Java System Instant Messaging 7 2005Q4	Direct upgrade: Performed by applying patches.	None except for enabling new `httpbind` gateway feature and redirect server if desired.
Java Enterprise System 2005Q1	Sun Java System Instant Messaging 7 2005Q1	Direct upgrade: Performed by applying patches.	None except for enabling new `httpbind` gateway feature and redirect server if desired.
Java Enterprise System 2004Q2	Sun Java System Instant Messaging 6 2004Q2	Direct upgrade: Performed using the `upgrade` utility.	Configuration data; reapply customizations
Java Enterprise System 2003Q4	Sun ONE Instant Messaging 6.1	Direct upgrade not certified, but can be performed using the `upgrade` utility.	Configuration data; reapply customizations
Previous Sun ONE releases	Sun ONE Instant Messaing 6.0	No direct upgrade	

Instant Messaging Data

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

Table 7-2 Instant Messaging Data Usage

Type of Data	Location	Usage

Configuration data:	Local configuration directory
	`/etc/`*InstantMessagng-base*`/default/config/iim.conf`
	`/etc/`*InstantMessagng-base*`/default/config/registration.properties`
		Configuration of Instant Messaging processes and registration attributes

Instant Messaging Server Resources	Local configuration directory *InstantMessaging-base*`/html`	customized client files downloaded by end users to launch the Messenger client.

Table 7-2 Instant Messaging Data Usage *(Continued)*

Type of Data	Location	Usage
Dynamic data	*InstantMessagingRuntime-base*/default/db	All variable data, such as the Instant Messaging database, log files, and lock files.

Instant Messaging Upgrade Strategy

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

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

Compatibility Issues

Instant Messaging 7.2 includes a new public interface (the httpbind gateway), however this represents a new feature that does not affect backward compatibility with earlier versions.

However, the change in Multiplexor protocol, a private interface, means that while a 7.2 Instant Messaging server can communicate with a 2005Q4 Multiplexor component, a 7.2 Instant Messaging Multiplexor component is unable to communicate with a 2005Q4 server. This limitation, impacts the order in which components need to be upgraded.

If you intend to use a redirect server in your deployment, you must also upgrade the client (Instant Messenger).

Instant Messaging Dependencies

Instant Messaging dependencies on other Communications Suite components can impact the procedure for upgrading and re-configuring Instant Messaging software. Changes in Instant Messaging interfaces or functions, for example, could require upgraded version of components upon which Instant Messaging depends. The need to upgrade such components depends upon the specific upgrade path.

Instant Messaging has dependencies on the following Communications Suite components:

- **Shared components.** Instant Messaging has dependencies on specific Communications Suite shared components (see Table 1-6 on page 31).
- **Web Container.** Instant Messaging clients depend upon web container services, which can be provided either by Communications Suite Web Server or Communications Suite Application Server.
- **Access Manager (optional).** For software solutions that support single user sign-on for web-based services, Instant Messaging can be configured to use Access Manager single sign-on capability.
- **Directory Server (optional).** Instant Messaging can be configured to store and access user data in Directory Server. As a result, Instant Messaging upgrades might require extensions of directory schema.

Web Container Upgrade Scenarios

Instant Messaging client resources can be deployed in a web container provided by either Web Server or Application Server. As a result, the upgrade of Instant Messaging to 7.2 can be complicated by the possibility of also having upgraded 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 7-3 Web Container Upgrade Scenarios for Instant Messaging Upgrade

Scenario	Web Container in which Instant Messaging is Originally Deployed	Web Container in which Instant Messaging is Deployed After Upgrade	Applicable Instant Messaging Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 6.*x*	2005Q1 2005Q4
Scenario 2	Web Server 6.*x*	Web Server 7.0	2004Q2 2005Q1 2005Q4
Scenario 3	Application Server 8.1	Application Server 8.1	2005Q1 2005Q4
Scenario 4	Application Server 8.1	Application Server 8.2	2005Q1 2005Q4
Scenario 5	Application Server 7*x*	Application Server 8.2	2004Q2

You must be careful when upgrading Instant Messaging (for example when re-deploying Instant Messaging resources) to provide values appropriate to the upgrade scenario in Table 7-3 that applies, especially when there is a major version upgrade of the web container.

Upgrading Instant Messaging from Java Enterprise System 2005Q4 or 2006Q1

This section includes the following information about upgrading Instant Messaging from Communications Suite 2005Q4 to Communications Suite 5. The section covers the following topics:

Introduction

When upgrading 2005Q4 or 2006Q1 Instant Messaging to 7.2, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the previous version, and explicitly re-deploying client resources. There is no re-configuration of Instant Messaging required, except to enable the httpbind gateway feature.
- **Upgrade Dependencies.** While Instant Messaging has dependencies on a number of Communications Suite shared components (see Table 1-6 on page 31), Instant Messaging 7.2 requires only that IM-SDK be upgraded to Communications Suite 5.

  In addition, Instant Messaging 7.2 is dependent on a web container and optionally dependent on Access Manager and Directory Server, as described in "Instant Messaging Dependencies" on page 138. These are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Instant Messaging to 7.2.
- **Backward Compatibility.** Instant Messaging 7.2 is backwardly compatible with the 2005Q4 and 2006Q1 versions.

- **Upgrade Rollback.** Rollback of the 7.2 upgrade to 2005Q4 or 2006Q1 is achieved on Solaris platforms by removing the patches applied during the upgrade and redeploying the client resources to the web container.
- **Platform Issues.** The general approach for upgrading Instant Messaging is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

2005Q4 and 2006Q1 Instant Messaging Upgrade

This section describes how to perform an upgrade of Instant Messaging from Communications Suite 2005Q4 or 2006Q1 to Communications Suite 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

Before you upgrade Instant Messaging you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Instant Messaging by entering the following command:

InstantMessaging-base`/sbin/imadmin version`

Table 7-4 Instant Messaging Version Verification Outputs

Release	Instant Messaging Version Number
Java Enterprise System 2004Q2	Version number `6.x`
Java Enterprise System 2005Q1	Version number `7.0`
Java Enterprise System 2005Q4	Version number `7.0.1`
Java Enterprise System 2006Q1	Version number `7.1-11`
Communications Suite 5	Version number `7.2-`*xx*

Upgrade Instant Messaging Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 5. However, Instant Messaging has a hard upgrade dependency only on the IM-SDK shared component.

If you choose to upgrade all Instant Messaging dependencies, they should be upgraded in the following order, all before you upgrade Instant Messaging. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for updating all Communications Suite shared components to Communications Suite 5 are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41. You must upgrade shared components before upgrading Instant Messaging.
2. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Access Manager (optional).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.
4. **Directory Server (optional).** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.

Back Up Instant Messaging Data

The Instant Messaging upgrade from 2005Q4 or 2006Q1 to Communications Suite 5 does not modify configuration data. However, as a safety measure, in case you re-configure something by accident, it is a good idea to back up the Instant Messaging database and any existing resource and configuration files before upgrading Instant Messaging. This would include, for example, the `iim.conf` file (for location, see Table 7-2 on page 137).

For more information, see the *Sun Java System Instant Messaging 7 2005Q4 Administration Guide,* `http://docs.sun.com/doc/819-2503`.

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. Depending on the web container you are using, you might need the corresponding administrator user ID and password.

Upgrading 2005Q4 or 2006Q1 Instant Messaging (Solaris)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by the procedure itself.

Upgrade Considerations (Solaris)

The upgrade of Instant Messaging software to Communications Suite 5 takes into account the following considerations:

- Instant Messaging components should be shut down when patches are being applied to the installed image.
- In deployment architectures in which different Instant Messaging subcomponents reside on different computers, for example messenger resources on one computer, Instant Messaging Server on another, and Instant Messaging Multiplexor on yet another, the upgrade must be performed on all such computers.
- Because of a change in Multiplexor protocol, an Instant Messaging 7.2 server component can communicate with a 2005Q4 or 2006Q1 Instant Messaging Multiplexor, but not the other way around. Therefore in distributed deployments, you should upgrade the server components before upgrading the Multiplexor components.
- The Instant Messaging upgrade applies to multiple subcomponents of Instant Messaging on one computer using the same installed image.
- The Instant Messaging upgrade patches for Solaris OS are shown in the following table. Patch revision numbers are the minimum required for upgrade to Communications Suite 5. If newer revisions become available, use the newer ones instead of those shown in the table.

Table 7-5 Patches to Upgrade Instant Messaging on Solaris

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Instant Messaging core	118786-26	118787-26
Instant Messaging localization	120841-02	120841-02

Upgrade Procedure (Solaris)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 7-5.

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

2. Log in as root or become superuser.

   `su -`

3. Stop Instant Messaging if it is running.

   InstantMessaging-base`/sbin/imadmin stop`

4. If you have not already done so, update all shared component to Communications Suite 5.

   See "Upgrade Instant Messaging Dependencies" on page 142.

5. Apply the appropriate Instant Messaging patches in Table 7-5.

   If needed, be sure to apply the Instant Messaging localization patch before applying the core patch.

   `patchadd` *patch_ID*

6. Confirm that the patch upgrades were successful:

   `showrev -p | grep` *patch_ID*

   The output should return the versions of patch IDs applied in Step 5.

7. Re-deploy Instant Messaging client resources in a web container instance.

   Web container upgrade scenarios 1, 3, and 4:

   If the web container had not been upgraded in a pre-upgrade task (Scenarios 1 and 3 of Table 7-3 on page 139) or if the web container had been upgraded so that instance configuration data remained in a known location (Scenario 4), then the re-deployment of client resources is performed as follows:

   a. Run the configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

      Do not re-configure server components.

c. Select the Deploy Messenger Resources download checkbox.

The `configure` utility requests necessary web container information, queries the web container for configuration information, generates `.war` files, and deploys them.

d. *InstantMessaging-base*`/sbin/iwadmin redeploy im`

You are prompted for any required admin ID and password.

e. Restart the Instant Messaging service that was stopped in Step 3.

InstantMessaging-base`/sbin/imadmin start`

Web container upgrade scenario 2:

However, if the web container had been upgraded so that instance configuration data moved to a new location (Scenario 2 of Table 7-3 on page 139) then the re-deployment of client resources is performed using the Instant Messaging `configure` utility, as follows:

a. Run the configurator program.

InstantMessaging-base`/configure`

b. In the interface, select only Client Components.

Do not re-configure server components.

c. Select the Deploy Messenger Resources download checkbox.

The `configure` utility requests necessary web container information, queries the web container for configuration information, generates `.war` files, and deploys them.

d. Restart the Instant Messaging service that was stopped in Step 3.

InstantMessaging-base`/sbin/imadmin start`

Upgrading 2005Q4 or 2006Q1 Instant Messaging (Linux)

This section discusses considerations that impact the upgrade procedure for Instant Messaging followed by the procedure.

Upgrade Considerations (Linux)

The upgrade of Instant Messaging software to Communications Suite 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 143), except that the Linux 5 upgrade patches differ from the Solaris patches.

The Instant Messaging upgrade patches for Linux OS are shown in the following table. Patch revision numbers are the minimum required for upgrade to Communications Suite 5. If newer revisions become available, use the newer ones instead of those shown in the table

Table 7-6 Patches to Upgrade Instant Messaging on Linux

Description	Patch ID and RPM names
Instant Messaging core	118788-26 • `sun-im-client-7.2-24.i386.rpm` • `sun-im-server-7.2-24.i386.rpm` • `sun-im-mux-7.2-24.i386.rpm` • `sun-im-olh-7.2-24.i386.rpm` • `sun-im-ident-7.2-24.i386.rpm` • `sun-im-install-7.2-24.i386.rpm` • `sun-im-apidoc-7.2-24.i386.rpm`
Instant Messaging localization	120842-2 • `sun-im-client-`*`Locale`*`-7.2-9.i386.rpm` • `sun-im-ident-`*`Locale`*`-7.2-9.i386.rpm` • `sun-im-install-`*`Locale`*`-7.2-9.i386.rpm` • `sun-im-olh-`*`Locale`*`-7.2-9.i386.rpm`

Upgrade Procedure (Linux)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

CAUTION An upgrade from Communications Suite 2005Q4 or 2006Q1 to Communications Suite 5 on Linux cannot be rolled back.

Also, you must apply the `sun-im-ident` RPM before you apply the `sun-im-install` RPM. If you do not, upgrade will fail.

1. Obtain the required patch using the patch number and RPM name from Table 7-6. Use this information to obtain the version numbers for the RPM.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. Stop Instant Messaging if it is running.

   InstantMessaging-base`/sbin/imadmin stop`

4. If you have not already done so, synchronize all shared component to Communications Suite 5.

   See "Upgrade Instant Messaging Dependencies" on page 142.

5. Apply the RPMs for Instant Messaging in Table 7-6.

   If needed, be sure to apply the Instant Messaging localization patch before applying the Instant Messaging core patch.

   ```
   rpm -Fvh sun-im-module-Locale-7.2-9.i386.rpm
   rpm -Fvh sun-im-module-7.2-9.i386.rpm
   ```

6. Confirm that the patch upgrades were successful:

   ```
   rpm -qa | grep sun-im
   ```

   The new version numbers of the RPMs should be returned.

7. Re-deploy Instant Messaging client resources in a web container instance.

   Web container upgrade scenarios 1, 3, and 4:

   If the web container had not been upgraded in a pre-upgrade task (Scenarios 1 and 3 of Table 7-3 on page 139) or if the web container had been upgraded so that instance configuration data remained in a known location (Scenario 4), then the re-deployment of client resources is performed as follows:

   a. Run the configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

      Do not re-configure server components.

   c. Select the Deploy Messenger Resources download checkbox.

      The `configure` utility requests necessary web container information, queries the web container for configuration information, generates `.war` files, and deploys them.

d. *InstantMessaging-base*/sbin/iwadmin redeploy im

You are prompted for any required admin ID and password.

e. Restart the Instant Messaging service that was stopped in Step 3.

InstantMessaging-base/sbin/imadmin start

Web container upgrade scenario 2:

However, if the web container had been upgraded so that instance configuration data moved to a new location (Scenario 2 of Table 7-3 on page 139) then the re-deployment of client resources is performed using the Instant Messaging configure utility, as follows:

a. Run the configurator program.

InstantMessaging-base/configure

b. In the interface, select only Client Components.

Do not re-configure server components.

c. Select the Deploy Messenger Resources download checkbox.

The configure utility requests necessary web container information, queries the web container for configuration information, generates .war files, and deploys them.

d. Restart the Instant Messaging service that was stopped in Step 3.

InstantMessaging-base/sbin/imadmin start

Verifying the Upgrade

You can verify the upgrade of Instant Messaging by entering the following command:

InstantMessaging-base/sbin/imadmin version

See Table 7-4 on page 142 for output values.

You can also check the status of the various Instant Messaging subcomponents using the following command:

InstantMessaging-base/sbin/imadmin status

Or you can check the log file located at *iim.instancevardir*/log,

where *instancevardir* is specified in the iim.conf file (for path, see Table 7-2 on page 137).

To confirm that the upgrade is properly localized:

1. Launch localized Instant Messaging home page.

   http://*hostName*:*port*/uri/*Locale*

2. On localized Instant Messaging home page, launch Instant Messaging client login dialog by clicking Java Web Start button or Java Plugin button.

3. Enter your user name and password on localized Instant Messaging client and make sure that localized client is displayed.

Post-Upgrade Tasks

If you want to enable the `httpbind` gateway as part of your upgrade to 5, you need to perform the following steps:

1. Configure the httpbind component in the server.

   See the *Sun Java System Instant Messaging 7.2 Administration Guide,* http://docs.sun.com/doc/819-4412.

2. Deploy the `httpbind` gateway in a web container instance.

   a. Run the Instant Messaging configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

   c. De-select the Deploy Messenger Resources download checkbox.

   d. Select the Deploy `httpbind` Gateway download checkbox.

Rolling Back the Upgrade (Solaris)

This section describes considerations that impact the upgrade rollback procedure for Instant Messaging followed by the procedure itself.

Rollback Considerations (Solaris)

The procedure for rolling back the upgrade from Instant Messaging 7.2 to 2006Q1 or 2005Q4 is almost the reverse of the procedure for upgrading.

Rollback Procedure (Solaris)

1. Log in as root or become superuser.

   ```
   su -
   ```

2. Stop Instant Messaging if it is running.

   InstantMessaging-base`/sbin/imadmin stop`

3. Remove the patches in Table 7-5 on page 143.

   `patchrm` *patch_ID*

   The patch removal will undeploy the current client resources and redeploy the older version.

4. Manually remove the 7.2 schema from your Directory Server and then apply the schema from the version to which you rolled back.

   This includes removing sunimuser and sunpresenceuser object classes from all users in your directory.

5. If you had run the Instant Messaging 7.2 configurator program to configure or deploy client resources, you must run the 2005Q4 or 2006Q1 configurator program to re-configure or re-deploy the client resources.

6. Restart the Instant Messaging service that was stopped in Step 2.

   InstantMessaging-base`/sbin/imadmin start`

Multiple Instance Upgrades

If you have multiple Instant Messaging Server instances using the same installed image, upgrading the default instance will upgrade all other instances. However, only the client components of the *default* instance will be redeployed as part of the upgrade.

The web-based client components corresponding to the additional (non-default) Instant Messaging Server instances, however, have to be manually redeployed to the web container as follows:

1. Reuse the working directory, or a copy thereof, created when the additional instance was first created.

   The steps for creating an additional instance are described in the *Sun Java System Instant Messaging 7.2 Administration Guide,* `http://docs.sun.com/doc/819-4412`..

2. In the working directory, replace only the non-customized files, that is, the contents of the upgraded program files (the .jar file) for this instance.

3. Generate a `.war` file from the working directory.

4. Un-deploy the 2005Q4 web application corresponding to this instance.

5. Deploy the 7.2 `.war` file generated in Step 3.

   You will have to use the web container's deployment interface.

Upgrading Instant Messaging from Java Enterprise System 2005Q1

On Linux, you must uninstall the `sun-im-ident-7.0-13` package before upgrading. Otherwise, the procedure for upgrading Communications Suite 2005Q1 Instant Messaging to 7.2 is the same as that for upgrading 2005Q4 or 2006Q1 Instant Messaging to 7.2 regardless of operating system.

To upgrade 2005Q1 Instant Messaging to 7.2, use the instructions in "Upgrading Instant Messaging from Java Enterprise System 2005Q4 or 2006Q1" on page 140, except substitute 2005Q1 wherever 2005Q4 is referenced.

Upgrading Instant Messaging from Java Enterprise System 2004Q2

This section includes information about upgrading Instant Messaging from Java Enterprise System 2004Q2 to Communications Suite 5 (Instant Messaging 7.2). The procedure for upgrading 2004Q2 Instant Messaging to 7.2 is quite different from that for upgrading from 2005Q4 Instant Messaging.

The section covers the following topics:

Introduction

When upgrading Communications Suite 2004Q2 Instant Messaging to 5, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed using an upgrade utility that performs all operations needed to upgrade Instant Messaging software.

- **Upgrade Dependencies.** Upgrade of any Communications Suite component on a computer from 2004Q2 requires the upgrade of all other Communications Suite components hosted by the computer; selective upgrade of Communications Suite components from 2004Q2 to 7.2 is not supported. In particular, all Communications Suite shared components used by Instant Messaging, the web container, and Access Manager need to be upgraded to Communications Suite 5.
- **Backward Compatibility.** Instant Messaging 7.2 is backwardly compatible with the 2004Q2 version.
- **Upgrade Rollback.** Rollback of the 7.2 upgrade to 2004Q2 is achieved by saving all 2004Q2 software and data and manually reverting back to the 2004Q2 version. There is no utility for rolling back the upgrade.
- **Platform Issues.** The general approach for upgrading Instant Messaging is the same on both Solaris and Linux operating systems. The upgrade process includes any platform-specific details.

2004Q2 Instant Messaging Upgrade

This section describes how to perform an upgrade of Instant Messaging from Java Enterprise System 2004Q2 to Communications Suite 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

Before you upgrade Instant Messaging you should perform the tasks described below.

Verify Current Version Information

You can verify the current version of Instant Messaging using standard version checking utilities:

Solaris:

```
pkginfo -l SUNWiimin
```

Linux:

```
rpm -qa | grep sun-im
```

See Table 7-4 on page 142 for output values.

Upgrade Instant Messaging Dependencies

Communications Suite 5 does not support the coexistence of Instant Messaging 2004Q2 and Instant Messaging 7.2 shared components on a single computer.

You are therefore required to upgrade all local Communications Suite 2004Q2 components on which Instant Messaging depends to Communications Suite 5. When you upgrade all Instant Messaging dependencies on a computer, they should be upgraded in the following order, all before you upgrade Instant Messaging.

1. **Shared Components.** All shared components must be updated to 5. Instructions for updating Communications Suite shared components to 5 are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41).
2. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager to are provided in the *Java Enterprise System Upgrade Guide*.
4. **Directory Server (optional).** Instant Messaging is rarely dependent on a local Directory Server, however, instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.

Back Up Instant Messaging Data

The Instant Messaging upgrade from 2004Q2 to 7.2 modifies configuration data and customizations. Before upgrading Instant Messaging, it is a good idea to back up the Instant Messaging database and any existing resource and configuration files. For more information, see the *Sun Java System Instant Messaging 7 2005Q4 Administration Guide,* `http://docs.sun.com/doc/819-2503`.

Obtain Required Configuration Information and Passwords

You have to log in as superuser to perform the upgrade. Depending on the web container you are using, you might need the corresponding administrator user ID and password.

Upgrading 2004Q2 Instant Messaging (Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Instant Messaging software to Communications Suite 5 takes into account the following considerations:

- In deployment architectures in which different Instant Messaging subcomponents reside on different computers, for example messenger resources on one computer, Instant Messaging Server on another, and Instant Messaging Multiplexor on yet another, the upgrade must be performed on all such computers.
- Because of a change in Multiplexor protocol, a Instant Messaging 7.2 server component can communicate with a 2005Q4 Instant Messaging Multiplexor, but not the other way around. Therefore in distributed deployments, you should upgrade the server components before upgrading the Multiplexor components.
- The Instant Messaging upgrade applies to multiple subcomponents of Instant Messaging on one computer using the same installed image.
- The upgrade of 2004Q2 Instant Messaging software to Communications Suite 5 makes use of an `upgrade` utility that performs the following operations:
  - Creates a temporary directory where it stores working files.
  - Gathers and temporarily stores previous package installation parameters for all packages installed on the system.
  - Shuts down the previous version of the Instant Messaging server.
  - Installs new packages and patches existing packages.
  - Installs any new shared component packages needed by Instant Messaging.
  - Saves the previous graphics contents from IIM_DOCROOT and restores them to the new resource files location.
  - Restarts all services.
  - Deletes the temporary directory and its contents.

Upgrade Procedure (Solaris)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, upgrade the IM-SDK shared component and all other shared components to 5.

   See "Upgrade Instant Messaging Dependencies" on page 153.

3. Run the `upgrade` utility from the Instant Messaging tools directory of the Communications Suite 5 distribution.

   ```
   cd /Solaris_arch/Product/instant_messaging/Tools
   ./upgrade
   ```

   The `upgrade` utility creates a log file that shows the progression of the upgrade process. The log file resides in the following location:

   `/var/sadm/install/logs/Instant_Messaging_Upgrade.`*timestamp*

   Where *timestamp* is in the format *yyyymmddhhss*.

4. Re-configure Instant Messaging.

   Change configuration as necessary to use new features introduced after 2004Q2.

   InstantMessaging-base`/configure`

   If you *have not* customized messenger resources, choose to configure both client and server components when prompted. This completes the upgrade procedure.

   If you *have* customized messenger resources, choose to configure only the server when prompted, then complete Step 5 and Step 6 that follow. For configuration information, see *Sun Java System Instant Messaging 7.2 Administration Guide,* `http://docs.sun.com/doc/819-4412`..

5. Re-customize messenger resources.

   If you had customized your messenger resources, you need to reapply those customizations to the following files:

   InstantMessaging-base`/html/`*Locale*`/imbrand.jar`
   InstantMessaging-base`/html/`*Locale*`/imb[ssl].html|jnlp`

   Consult the customized files that you saved under "Back Up Instant Messaging Data" on page 153.

6. If you customized messenger resources, re-deploy the resources in a web container instance.

   Because the web container had been upgraded as a pre-upgrade task (Scenarios 2 and 5 of Table 7-3 on page 139), then the re-deployment of client resources is performed using the Instant Messaging configurator program, as follows:

   a. Run the configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

      Do not re-configure server components.

   c. Select the Deploy Messenger Resources download checkbox.

      The `configure` program requests necessary web container information, generates `war` files, and deploys them.

Upgrading 2004Q2 Instant Messaging (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Instant Messaging software to Communications Suite 5 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 154).

Upgrade Procedure (Linux)

The procedure documented below applies to Instant Messaging on the computer where the upgrade is taking place.

1. Log in as root or become superuser.

   ```
   su -
   ```

2. If you have not already done so, upgrade the IM-SDK shared component to 5 and any other shared components you wish to upgrade.

   See "Upgrade Instant Messaging Dependencies" on page 153.

3. Run the `upgrade` utility from the Instant Messaging tools directory of the Communications Suite 5 distribution.

   ```
   cd /Linux_x86/Product/instant_messaging/Tools
   ./upgrade
   ```

The `upgrade` utility creates a log file that shows the progression of the upgrade process. The log file resides in the following location:

`/var/sadm/install/logs/Instant_Messaging_Upgrade.`*timestamp*

Where *timestamp* is in the format *yyyymmddhhss*.

4. Re-configure Instant Messaging.

   Change configuration as necessary to use new features introduced after 2004Q2.

   InstantMessaging-base`/configure`

   If you have not customized messenger resources, choose to configure both client and server components when prompted. If you have customized messenger resources, choose to configure only the server when prompted. For configuration information, see the *Sun Java System Instant Messaging 7.2 Administration Guide,* `http://docs.sun.com/doc/819-4412`..

5. Re-customize messenger resources.

   If you had customized your messenger resources, you need to reapply those customizations to the following files:

   InstantMessaging-base`/html/`*Locale*`/imbrand.jar`

   InstantMessaging-base`/html/`*Locale*`/imb[ssl].html|jnlp`

   Consult the customized files that you saved under "Back Up Instant Messaging Data" on page 153.

6. If you customized messenger resources, re-deploy the resources in a web container instance.

   Because the web container had been upgraded as a pre-upgrade task (Scenarios 2 and 5 of Table 7-3 on page 139), then the re-deployment of client resources is performed using the Instant Messaging configurator program, as follows:

   a. Run the configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

      Do not re-configure server components.

   c. Select the Deploy Messenger Resources download checkbox.

      The `configure` program requests necessary web container information, generates `war` files, and deploys them.

Verifying the Upgrade

You can verify the upgrade of Instant Messaging to 5 by entering the following command:

InstantMessaging-base`/sbin/imadmin version`

See Table 7-4 on page 142 for output values.

You can also check the status of the various Instant Messaging subcomponents using the following command:

InstantMessaging-base`/sbin/imadmin status`

Or you can check the log file located at *iim.instancevardir*/log,

where *instancevardir* is specified in the `iim.conf` file (for path, see Table 7-2 on page 137).

To confirm that the upgrade is properly localized:

1. Launch localized Instant Messaging home page.

   http://*hostName*:*port*/uri/*Locale*

2. On localized Instant Messaging home page, launch Instant Messaging client login dialog by clicking Java Web Start button or Java Plugin button.

3. Enter your user name and password on localized Instant Messaging client and make sure that localized client is displayed.

Post-Upgrade Tasks

If you want to enable the `httpbind` gateway as part of your upgrade to 5, you need to perform the following steps:

1. Configure the httpbind component in the server.

   See the *Sun Java System Instant Messaging 7.2 Administration Guide,* `http://docs.sun.com/doc/819-4412`..

2. Deploy the `httpbind` gateway in a web container instance.

   a. Run the Instant Messaging configurator program.

      InstantMessaging-base`/configure`

   b. In the interface, select only Client Components.

   c. Un-select the Deploy Messenger Resources download.

   d. Select the Deploy `httpbind` Gateway download.

Rolling Back the Upgrade

Rollback of the 5 upgrade to 2004Q2 is achieved by saving all 2004Q2 software and data (see "Back Up Instant Messaging Data" on page 153) and manually reverting back to the 2004Q2 version. There is no utility for rolling back the upgrade.

Multiple Instance Upgrades

If you have multiple Instant Messaging Server instances using the same installed image, upgrading the default instance will upgrade all other instances. However, only the client components of the *default* instance will be redeployed as part of the upgrade.

The web-based client components corresponding to the additional (non-default) Instant Messaging Server instances, however, have to be manually redeployed to the web container as follows:

1. Reuse the working directory, or a copy thereof, created when the additional instance was first created.

   The steps for creating an additional instance are described in the *Sun Java System Instant Messaging 7.2 Administration Guide,* `http://docs.sun.com/doc/819-4412`.

2. In the working directory, replace only the non-customized files, that is, the contents of the upgraded program files (the .jar file) for this instance.
3. Generate a `war` file from the working directory.
4. Un-deploy the 2005Q4 web application corresponding to this instance.
5. Deploy the 5 `war` file generated in Step 3.

   You will have to use the web container's deployment interface.

Upgrading Instant Messaging from the Interim Feature Release 2006Q1 (7.1)

The procedure for upgrading the Instant Messaging Interim Feature Release 2006Q1 (7.1) to 5 is the same as that for upgrading 2005Q4 Instant Messaging to 5.

To upgrade the Instant Messaging 2006Q1 to 5, use the instructions in "Upgrading Instant Messaging from Java Enterprise System 2005Q4 or 2006Q1" on page 140, except substitute 2006Q1 wherever 2005Q4 is referenced.

Chapter 8

Upgrading Delegated Administrator

This chapter describes how to upgrade Sun Java System Delegated Administrator to version 6.4 (Communications Suite 5). The chapter provides an overview of upgrade considerations for the different upgrade paths supported by version 6.4. The chapter covers upgrades on both the Solaris and Linux operating systems:

NOTE File locations in this chapter are specified with respect to two directory paths referred to as *DelegatedAdmin-base* and *DelegatedAdminConfig-base*.

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

Path Name	Solaris OS	Linux OS
DelegatedAdmin-base	`/opt/SUNWcomm`	`/opt/sun/comms/commcli`
DelegatedAdminConfig-base	`/var/opt/SUNWcomm`	`/var/opt/sun/comms/commcli`

Overview of Delegated Administrator Upgrades

This section describes the following general aspects of Delegated Administrator that impact upgrading to version 6.4:

About Delegated Administrator

Delegated Administrator represents a minor update with respect to 2005Q4. The only new feature, along with bug fixes, is support for calendar group provisioning.

Delegated Administrator Upgrade Roadmap

Table 8-1 shows the supported Delegated Administrator upgrade paths to version 6.4. The table applies to both Solaris and Linux operating systems.

Table 8-1 Upgrade Paths to Communications Suite 5: Sun Java System Delegated Administrator 6.4

Communications Suite Release	Delegated Administrator Version	General Approach	Re-configuration Required
2005Q4	Sun Java System Communication Services Delegated Administrator 6.3 2005Q4	Direct upgrade: Performed by applying patches and running `config-commda` utility.	Configuration data
2005Q1	Sun Java System Communication Services Delegated Administrator 6 2005Q1	Direct upgrade: Performed by applying patches and running `config-commda` utility.	Configuration data
2004Q2	Sun Java System Communication Services User Management Utility 1.1 (2004Q2)	Direct upgrade: Performed by applying patches and running `config-commda` utility.	Configuration data

Table 8-1 Upgrade Paths to Communications Suite 5: Sun Java System Delegated Administrator 6.4 *(Continued)*

Communications Suite Release	Delegated Administrator Version	General Approach	Re-configuration Required
2003Q4	User Management Utility (2003Q4)	Direct upgrade not certified: But can be performed by applying patches and running `config-commda` utility.	Configuration data
Pre-dates Communications Suite releases	iPlanet Delegated Administrator	No direct upgrade	

Delegated Administrator Data

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

Table 8-2 Delegated Administrator Data Usage

Type of Data	Location	Usage
Directory schema	Directory Server user/group directory	For attributes needed to support end users, organizations, and services schema
Delegated Administrator variable data	*DelegatedAdminConfig-base*	Silent install `saveState` files and so forth.

Table 8-2 Delegated Administrator Data Usage *(Continued)*

Type of Data	Location	Usage
Web container access control and configuration files	Web Server 7.0 (Java Enterprise System 5) `server.policy` and `server.xml` files in *WebServer7Config-base*`/https-`*configName*`/config`	Configuration of Delegated Administrator web container instance
	Web Server 6.*x* (Java Enterprise System 2004Q2, 2005Q1, and 2005Q4) `server.policy` and `server.xml` files in *WebServer6-base*`/https-`*hostname*`/config`	
	Application Server 8.*x* (Java Enterprise System 2005Q1, 2005Q4, and 5): `server.policy` and `domain.xml` files in *AppServer8Config-base*`/domains/`*domainName*`/config`	
	Application Server 7.*x* (Java Enterprise System 2004Q2): `server.policy` and `server.xml` files in *AppServer7Config-base*`/domains/`*domainName*`/config`	
Configuration data (2005Q1 and 2005Q4)	*DelegatedAdmin-base*`/data/WEB-INF/classes/sun/comm/cli/server/servlet/resource.properties`	Delegated Administrator server configuration and customizations
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/daconfig.properties`	Delegated Administrator console configuration
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/logger.properties`	Delegated Administrator console logging
	DelegatedAdmin-base`/data/da/WEB-INF/classes/com/sun/comm/da/resources/security.properties`	Delegated Administrator console configuration
Configuration data (6.4)**	…`/commcli/WEB-INF/classes/sun/comm/cli/server/servlet/resource.properties`[1]	Delegated Administrator server configuration and customizations
	…`/da/WEB-INF/classes/com/sun/comm/da/resources/daconfig.properties`[1]	Delegated Administrator console configuration
	…`/da/WEB-INF/classes/com/sun/comm/da/resources/logger.properties`[1]	Delegated Administrator console logging
	…`/da/WEB-INF/classes/com/sun/comm/da/resources/Security.properties`[1]	Delegated Administrator console security configuration

1. The unspecified part of the path is the location of the corresponding web application directory of the Web Server or Application Server instance providing the web container.

**For information about how to deploy the Delegated Administrator 6.4 configuration data to the web container directories, see “Deploying a Customized Configuration File” in “Chapter 4: Customizing Delegated Administrator” in the *Sun Java System Delegated Administrator 6.4 Administration Guide*.

Delegated Administrator Upgrade Strategy

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

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

Compatibility Issues

Delegated Administrator 6.4 introduces no changes in public interfaces and is backwardly compatible with earlier versions.

Delegated Administrator Dependencies

Delegated Administrator dependencies on other Communications Suite components can impact the procedure for upgrading and re-configuring Delegated Administrator software. Changes in Delegated Administrator interfaces or functions, for example, could require upgraded version of components upon which Delegated Administrator depends. The need to upgrade such components depends upon the specific upgrade path.

Delegated Administrator has dependencies on the following Communications Suite components:

- **Shared components.** Delegated Administrator has dependencies on specific Communications Suite shared components (see Table 1-6 on page 31).
- **Web Container.** Delegated Administrator depends upon web container services, which can be provided either by Sun Java System Web Server or Sun Java System Application Server.
- **Directory Server.** Delegated Administrator stores application and user data in Directory Server.
- **Directory Preparation Tool.** Delegated Administrator uses the Directory Preparation Tool to prepare the directory to support Delegated Administrator user provisioning functions. As a result, Delegated Administrator upgrades might depend upon preparation of the directory to support new functions.
- **Access Manager (Access Manager SDK).** Delegated Administrator depends upon Access Manager to register services and to make entries into Directory Server.

Web Container Upgrade Scenarios

Delegated Administrator can be deployed in a web container provided by either Web Server or Application Server. As a result, the upgrade of Delegated Administrator to version 6.4 can be complicated by the possibility of also having upgraded to Java ES 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 8-3 Web Container Upgrade Scenarios for Delegated Administrator Upgrade

Scenario	Web Container in which Delegated Administrator is Originally Deployed	Web Container in which Delegated Administrator is Deployed After Upgrade	Applicable Delegated Administrator Upgrade Paths: Upgrades From
Scenario 1	Web Server 6.*x*	Web Server 6.*x*	2005Q1 2005Q4
Scenario 2	Web Server 6.*x*	Web Server 7.0	2004Q2 2005Q1 2005Q4
Scenario 3	Application Server 8.1	Application Server 8.1	2005Q1 2005Q4
Scenario 4	Application Server 8.1	Application Server 8.2	2005Q1 2005Q4
Scenario 5	Application Server 7*x*	Application Server 8.2	2004Q2

You must be careful when upgrading Delegated Administrator (for example when re-configuring or re-deploying Delegated Administrator) to provide values appropriate to the upgrade scenario that applies, especially when there is a major version upgrade of the web container.

This consideration applies particularly to Scenario 5, when you upgrade Application Server 7.*x* to Application Server 8.2, and you upgrade Delegated Administrator from 2004Q2 to 6.4. In Scenario 5, you must supply the following values when you run the Delegated Administrator configuration program, `config-commda`:

Application Server target name: `server1`
Application Server virtual server identifier: `server`

For details, see "Upgrading 2004Q2 Delegated Administrator: Application Server Web Container" on page 180

Upgrading Delegated Administrator from Communications Services 2005Q4

This section includes information about upgrading Delegated Administrator from Communications Services 2005Q4 to version 6.4. The section covers the following topics:

Introduction

When upgrading Communications Services 2005Q4 Delegated Administrator to version 6.4, consider the following aspects of the upgrade process:

- **General Upgrade Approach.** The upgrade is performed by applying patches to the 2005Q4 version. Re-configuration of Delegated Administrator is achieved by running the `config-commda` configuration utility.
- **Upgrade Dependencies.** While Delegated Administrator has dependencies on a number of Communications Suite shared components (see Table 1-6 on page 31), Delegated Administrator 6.4 is compatible with the 2005Q4 versions of these components. Upgrade of these shared components is therefore optional with respect to upgrade of Delegated Administrator to version 6.4.

  In addition, Delegated Administrator 6.4 is dependent upon a web container and on Access Manager, as described in "Delegated Administrator Dependencies" on page 165. These are soft upgrade dependencies; upgrade of these components is optional with respect to upgrade of Delegated Administrator to 5. (However, if Access Manager is upgraded, then Delegated Administrator must be upgraded also.)

  In addition, Delegated Administrator 6.4 has a hard upgrade dependency on Directory Preparation Tool; Communications Suite 5 Directory Preparation Tool is required to prepare Directory Server for user provisioning operations.
- **Backward Compatibility.** Delegated Administrator 6.4 is backwardly compatible with the 2005Q4 version.
- **Upgrade Rollback.** Rollback of the 6.4 upgrade of Delegated Administrator to 2005Q4 is not supported.
- **Platform Issues.** The general approach for upgrading Delegated Administrator is the same on both Solaris and Linux operating systems, however the patching technologies are different. The upgrade process therefore includes platform-specific procedures.

2005Q4 Delegated Administrator Upgrade

This section describes how to perform an upgrade of Delegated Administrator from 2005Q4 to version 6.4 on both the Solaris and Linux platform. Where a topic depends on platform-specific procedures, the topic will indicate the operating system to which it applies. The section covers the following topics:

Pre-Upgrade Tasks

Before you upgrade Delegated Administrator you should perform the tasks described below.

Back Up Delegated Administrator Data

The Delegated Administrator upgrade from 2005Q4 to version 6.4 requires re-configuration of Delegated Administrator. It is a good idea to back up configuration data as a safety precaution, and to back up any 2005Q4 graphical user interface customizations.

For information about preserving customized data from a previous version, see “If You Are Upgrading From a Previous Release of Delegated Administrator” in “Chapter 3: Configuring Delegated Administrator” in the *Sun Java System Delegated Administrator 6.4 Administration Guide*.

Verify Current Version Information

You can verify the current version of Delegated Administrator in the console:

1. Log in to the Delegated Administrator console using the top-level administrator user ID and password specified for the previous version of Delegated Administrator.

   `http://`*`hostName`*`:`*`port`*`/da/DA/Login`

   where *hostName*:*port* are values provided for the previous version of Delegated Administrator.

2. Click **Version** in the upper-left area of the Log-in page.

You can also verify the current version of the client component of Delegated Administrator (the Delegated Administrator command-line utility) by entering the following command:

DelegatedAdmin-base`/bin/commadmin -V`

Table 8-4 Delegated Administrator Version Verification Outputs

Release	Delegated Administrator Version Number
2004Q2	`User Management Utility 1.1`
2005Q1	`User Management Utility 6 2005Q1`
2005Q4	`Delegated Administrator 6.3-0.09`
Communications Suite 5	`Delegated Administrator 6.4-0.x`

Upgrade Delegated Administrator Dependencies

It is generally recommended that all Communications Suite components on a computer system (and in a computing environment) be upgraded to Communications Suite 5. However, Delegated Administrator has a hard upgrade dependency only on Directory Preparation Tool. Upgrading of other 2005Q4 components upon which Delegated Administrator depends is therefore optional.

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

1. **Shared Components.** Instructions for synchronizing **Communications Suite** shared components to **Communications Suite 5** are provided in "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in the *Java Enterprise System Upgrade Guide*. When you're upgrading from Web Server 6.x and 7.x, be sure to change the FQDN (Fully Qualified Domain Name) host name to `https-`*FQDNhostname* to comply with Web Server 7.x.
4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.

5. **Directory Preparation Tool.** Directory Preparation Tool needs to have been run against Directory Server before using Delegated Administrator 6.4. If Directory Preparation Tool has not already been run against Directory Server, upgrade Directory Preparation Tool to Communications Suite 5 and use it to modify and extend the schema of Directory Server (see Chapter 3, "Directory Preparation Tool.").

Obtain Required Configuration Information and Passwords

You should know the following information about your currently installed version:

- Access Manager administrator user ID and password
- Access Manager internal LDAP password
- Top level administrator user ID and password
- Web container administrator user ID and password, depending on your web container
- Web container host and port
- If you are using Sun Java System Web Server, you also need to know the Web Server's administration server host and port.

For a complete list of the information you should know, see "Gather Your Delegated Administrator Configuration Information" in "Chapter 2: Planning for Installation and Configuration" in the *Sun Java System Delegated Administrator 6.4 Administration Guide*.

Upgrading 2005Q4 Delegated Administrator (Solaris)

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

Upgrade Considerations (Solaris)

The upgrade of Delegated Administrator software to version 6.4 takes into account the following considerations:

- Delegated Administrator should not be used while patches are being applied to the installed image.
- When re-configuring Delegated Administrator, it should be deployed to the same web container as Access Manager.
- The Delegated Administrator 6.4 upgrade patch for Solaris OS is shown in the following table. Patch revision numbers are the minimum required for upgrade to 6.4. If newer revisions become available, use the newer ones instead of those shown in the table.

Table 8-5 Patches to Upgrade Delegated Administrator on Solaris

Description	SPARC Solaris 9 & 10	X86 Solaris 9 & 10
Delegated Administrator	121581-12	121582-12

Upgrade Procedure (Solaris)

The procedure documented below applies to Delegated Administrator on the computer where the upgrade is taking place.

1. Obtain the required patches, based on Table 8-5.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. If you have not already done so, synchronize all shared components to Communications Suite 5.

   See "Upgrade Delegated Administrator Dependencies" on page 169.

4. Apply the appropriate Delegated Administrator patches in Table 8-5.

   ```
   patchadd patch_ID
   ```

5. Confirm that the patch upgrade was successful:

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

   The output should return the versions of patch IDs applied in Step 4.

6. In the case of the following dependency upgrade scenario:

   - 2005Q1 or 2005Q4 Web Server has been upgraded to Java Enterprise System 5 Web Server (web container upgrade Scenario 2 in Table 8-3 on page 166)
   - 2005Q1 or 2005Q4 Access Manager has *not* been upgraded to Java Enterprise System 5 Access Manager.

   You must modify the *AccessManagerConfig-base*`/config/AMConfig.properties` file by replacing the following line:

   ```
   com.sun.identity.webcontainer=WEB_SERVER
   ```

   with:

   ```
   com.sun.identity.webcontainer=WS7
   ```

7. Re-configure Delegated Administrator.

CAUTION Once you run the `config-commda` configuration script, you cannot roll back an upgrade from 2005Q4 to version 6.4 on Solaris.

   a. Make sure Directory Server is running.

   b. Make sure the web container running Access Manager and Delegated Administrator is running.

      Web Server 6.*x:*
      WebServer6-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:*
To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

AppServer8-base/sbin/asadmin start-domain --user *admin_ID* --password *password domainName*

AppServer8-base/sbin/asadmin start-node-agent --startinstances=false --user *admin_ID* --password *password nodeagentName*

AppServer8-base/sbin/asadmin start-instance --user *admin_ID* --password *password instanceName*

In the above commands, the following conventions are used:

- *nodeAgentName* has the form *hostName*
- The default *domainName* is `domain1`
- The default *instanceName* is `server1`

where Access Manager and Delegated Administrator are deployed in a server instance in the *domainName* domain. The default *domainName* is `domain1` and the default server instance is `server1`.

c. Run the Delegated Administrator configuration script.

DelegatedAdmin-base`/sbin/config-commda`

I. Provide the location of the existing configuration data directory.

This is the value of the *DelegatedAdminConfig-base* path.

II. When prompted, choose option #2: Upgrade.

III. Provide the current configuration values.

The values provided in previous runs of the `config-commda` script can be found in the `saveState` file in the following location:

DelegatedAdminConfig-base`/setup/config-commda_`*YYYYMMDDHHMMSS*

IV. Enter values for the additional parameters in the following table:

In the case where Web Server has been upgraded to Java Enterprise System 5 (Scenario 2 in Table 8-3 on page 166), the default virtual server name value must be replaced by the actual virtual server name, which is the virtual server name migrated from 2005Q4. See the *Java Enterprise System Upgrade Guide* for more information.

Table 8-6 Information Requested by `config-commda` Script for Delegated Administrator Upgrade

Web Container	Parameter	Default Value
Web Server: Scenarios 1 & 2 in Table 8-3 on page 166	Virtual Server Identifier	Web Server 6.*x* `https-`*hostName.domainName* Web Server 7 *hostName.domainName*
	Web server instance name	Web Server 6.*x* `https-`*hostName.domainName* Web Server 7 *hostName.domainName*

Table 8-6 Information Requested by `config-commda` Script for Delegated Administrator Upgrade *(Continued)*

Web Container	Parameter	Default Value
Application Server: Scenarios 3 & 4 in Table 8-3 on page 166	Virtual Server Identifier	`server`
	App Server Target Name (This is the App Server instance name where you want Delegated Administrator to be deployed.)	`server`

For additional details of how to use the `config-commda` script, see Chapter 3, "Configuring Delegated Administrator," in the *Sun Java System Delegated Administrator 6.4 Administration Guide,* `http://docs.sun.com/doc/819-4438`.

8. In the case in which 2005Q1 or 2005Q4 Web Server has been upgraded to Java Enterprise System 5 Web Server (Scenario 2 in Table 8-3 on page 166), remove obsolete Web Server 6.*x* classpath entries.

   a. Locate the `<server-class-path>` element in the `server.xml` file migrated to the deployed instance of Web Server 7.0.

      The `server.xml` file is in the following directory:

      WebServer7 Config-base`/http-hostname/config/`

   b. Remove the following classpath entries:

      ```
      /opt/SUNWcomm/data/WEB-INF/classes;
      /opt/SUNWcomm/lib/jars/commcli-client.jar;
      /opt/SUNWcomm/lib/jars/jdapi.jar;
      ```

      (The required Delegated Administrator classpath is now appended to the JVM class-path-suffix.)

c. Use the Web Server 7.0 command-line utility, `wadm`, to bring the modified configuration in the Web Server 7.0 instance back into the Web Server 7.0 configuration store, so that the deployed instance and central configuration store remain in sync.

For example, type the following command:

WebServer7-base`/bin/wadm pull-config --config=`*configName nodehost*

where *configName* is the name of the Web Server 7.0 instance (typically, the host name where the Web Server 7.0 instance is running)

and *nodeHost* is the name of the host where the Web Server 7.0 central configuration has been deployed.

9. Stop and restart the Delegated Administrator web container.

Web Server 6.*x:*
WebServer6-base`/https-`*instanceName*`/stop`
WebServer6-base`/https-`*instanceName*`/start`

Web Server 7.0*:*
WebServer7Config-base`/https-`*configName*`/bin/stopserv`
WebServer7Config-base`/https-`*configName*`/bin/startserv`

Application Server *8.x:*
AppServer8-base`/bin/asadmin stop-domain --user` *admin_ID*
`--password` *password domainName*

To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

AppServer8-base`/sbin/asadmin start-domain --user` *admin_ID*
`--password` *password domainName*

AppServer8-base`/sbin/asadmin start-node-agent`
`--startinstances=false --user` *admin_ID* `--password` *password*
nodeagentName

AppServer8-base`/sbin/asadmin start-instance --user` *admin_ID*
`--password` *password instanceName*

In the above commands, the following conventions are used:

- *nodeAgentName* has the form *hostName*
- The default *domainName* is `domain1`
- The default *instanceName* is `server1`

Upgrading 2005Q4 Delegated Administrator (Linux)

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

Upgrade Considerations (Linux)

The upgrade of Delegated Administrator software to version 6.4 on the Linux platform takes into account the same considerations as on the Solaris platform (see "Upgrade Considerations (Solaris)" on page 170), except that the Linux 6.4 upgrade patches differ from the Solaris patches.

The Delegated Administrator 6.4 upgrade patch for Linux OS is shown in the following table. Patch revision numbers are the minimum required for upgrade to 6.4. If newer revisions become available, use the newer ones instead of those shown in the table.

Table 8-7 Patches to Upgrade Delegated Administrator on Linux

Description	Patch ID and RPM names
Delegated Administrator	121583-12 • `sun-commcli-client-1.1-12.xx.i386.rpm` • `sun-commcli-server-1.1-12.xx.i386.rpm`

Upgrade Procedure (Linux)

The procedure documented below applies to Delegated Administrator on the computer where the upgrade is taking place.

CAUTION An upgrade from 2005Q4 to version 6.4 on Linux cannot be rolled back.

1. Obtain the required patches using the patch numbers and RPM names from Table 8-7. Use this information to obtain the version numbers for the RPM.

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

2. Log in as root or become superuser.

   ```
   su -
   ```

3. If you have not already done so, upgrade all shared components, the web container, and Access Manager (or Access Manager SDK).

   See "Upgrade Delegated Administrator Dependencies" on page 169.

4. Apply the RPMs for Delegated Administrator in Table 8-7.

   ```
   rpm -Fvh sun-commcli-client-1.1-12.xx.i386.rpm
   rpm -Fvh sun-commcli-server-1.1-12.xx.i386.rpm
   ```

5. Confirm that the patch upgrade was successful:

   ```
   rpm -qa | grep sun-commcli
   ```

   The new version numbers of the RPMs should be returned.

6. Proceed, beginning with Step 6 on page 172 through Step 9 on page 176 in the procedure for "Upgrading 2005Q4 Delegated Administrator (Solaris)". Use these same steps to upgrade on Linux.

Verifying the Upgrade

You can verify successful upgrade of Delegated Administrator as follows:

1. Log in to the Delegated Administrator console using the top-level administrator user ID and password specified during re-configuration of Delegated Administrator.

   `http://`*hostName*`:`*port*`/da`

   where *hostName*`:`*port* are values provided during re-configuration of Delegated Administrator.

2. Check the version number by clicking **Version** in the upper-left area of the Log-in page.

   See Table 8-4 on page 169 for output values.

3. To verify the current version of the client component of Delegated Administrator (the Delegated Administrator command-line utility), enter the following command:

   DelegatedAdmin-base`/bin/commadmin -V`

Post-Upgrade Tasks

There are no post-upgrade tasks beyond the steps described in "Upgrade Procedure (Solaris)" on page 171 and "Upgrade Procedure (Linux)" on page 177.

Rolling Back the Upgrade

Rollback of Delegated Administrator is not supported. Changes made during the upgrade procedure, such as entries in Directory Server or in deploying Delegated Administrator into the web container cannot easily be backed out.

Upgrading Delegated Administrator from 2005Q1

The procedure for upgrading 2005Q1 Delegated Administrator to verison 6.4 is the same as that for upgrading 2005Q4 Delegated Administrator to 6.4.

To upgrade 2005Q1 Delegated Administrator to version 6.4, use the instructions in "Upgrading Delegated Administrator from Communications Services 2005Q4" on page 167, except substitute 2005Q1 wherever 2005Q4 is referenced.

Upgrading Delegated Administrator from 2004Q2

The procedure for upgrading 2004Q2 Delegated Administrator to version 6.4 is the same as that for upgrading 2005Q4 Delegated Administrator to 6.4, with a couple of exceptions, noted below.

Upgrade Delegated Administrator Dependencies

The pre-upgrade tasks for upgrading 2004Q2 Delegated Administrator to version 6.4 are similar to those for upgrading 2005Q4 Delegated Administrator to 6.4, with the exception that the upgrade of Delegated Administrator dependencies should include the synchronizing of all shared components to Communications Suite 5 (see Table 1-6 on page 31) and all locally-resident product components upon which Delegated Administrator depends.

When upgrading Delegated Administrator dependencies, they should be upgraded in the following order, all before you upgrade Delegated Administrator. You can skip any that might already have been upgraded.

1. **Shared Components.** Instructions for synchronizing **Communications Suite** shared components to **Communications Suite 5** are provided in Chapter 2, "Upgrading Communications Suite Shared Components" on page 41.
2. **Directory Server.** Instructions for upgrading Directory Server are provided in the *Java Enterprise System Upgrade Guide*.
3. **Web Container Software.** Instructions for upgrading Web Server or Application Server are provided in the *Java Enterprise System Upgrade Guide*.

4. **Access Manager (Access Manager SDK).** Instructions for upgrading Access Manager are provided in the *Java Enterprise System Upgrade Guide*.

5. **Directory Preparation Tool.** Directory Preparation Tool rarely resides on the same computer as Delegated Administrator, however, instructions for upgrading Directory Preparation Tool and running it against Directory Server are provided in Chapter 3, "Directory Preparation Tool."

2004Q2 Delegated Administrator Upgrade

The procedure for upgrading Delegated Administrator from 2004Q2 to version 6.4 depends on the web container in which you are deploying Delegated Administrator software.

Upgrading 2004Q2 Delegated Administrator: Web Server Web Container

To upgrade 2004Q2 Delegated Administrator to version 6.4, when deploying into a Web Server web container that has been upgraded to 5, follow the instructions in "Upgrading 2005Q4 Delegated Administrator (Solaris)" on page 170 or "Upgrading 2005Q4 Delegated Administrator (Linux)" on page 177, except substitute 2004Q2 wherever 2005Q4 is referenced.

Upgrading 2004Q2 Delegated Administrator: Application Server Web Container

To upgrade 2004Q2 Delegated Administrator to version 6.4, when deploying into an Application Server web container that has been upgraded to 5, you first follow the instructions in "Upgrading 2005Q4 Delegated Administrator (Solaris)" on page 170 or "Upgrading 2005Q4 Delegated Administrator (Linux)" on page 177, except substitute 2004Q2 wherever 2005Q4 is referenced.

The 2004Q2 Application Server instance in which Delegated Administrator was originally deployed, when upgraded to 5, was migrated under a node agent created by the upgrade process.

Upgrade of Delegated Administrator in this situation requires you to change Step 7 on page 172 as follows:

7. Re-configure Delegated Administrator.

a. Modify the *AccessManagerConfig-base*/`config/AMConfig.properties` file.

Replace the following line:

```
com.sun.identity.webcontainer=IAS7.0
```

with:

```
com.sun.identity.webcontainer=IAS8.1
```

b. Make sure the upgraded Application Server instance, in which Delegated Administrator is deployed, is running.

To start the Application Server instance, you start the Domain Administration Server (DAS), then the node agent under which the instance was migrated, and finally the instance itself using the following commands:

AppServer8-base`/sbin/asadmin start-domain --user` *admin_ID* `--password` *password domainName*

AppServer8-base`/sbin/asadmin start-node-agent --startinstances=false --user` *admin_ID* `--password` *password nodeagentName*

AppServer8-base`/sbin/asadmin start-instance --user` *admin_ID* `--password` *password instanceName*

In the above commands, and in subsequent steps, the following conventions are used:

- *nodeAgentName* has the form *hostName*
- The default *domainName* is `domain1`
- The default *instanceName* is `server1`

c. Undeploy the `commcli` Delegated Administrator web application from the Application Server instance.

AppServer8-base`/bin/asadmin undeploy --secure=false --user` *admin* `--target` *instanceName* `commcli`

If `asadmin` is running under SSL, set `secure=true`.

d. Run the Delegated Administrator configuration utility.

DelegatedAdmin-base`/sbin/config-commda`

For details of how to use this utility, see "Chapter 3, Configuring Delegated Administrator" of the *Sun Java System Communications Services 6 2005Q4 Delegated Administrator Guide,* `http://docs.sun.com/doc/819-2658`.

Specify the following parameters:

- The Access Manager host and port
- When asked where to deploy the Delegated Administrator console and Delegated Administrator server, specify the instance information for the upgraded Application Server instance where Access Manager is deployed.
- When asked to supply the Application Server configuration details, you must supply the following values for the target name and virtual server identifier:

  Target name: `server1`

  Virtual server identifier: `server`

  You must specify these values because the asupgrade utility migrates the Application Server 7 server1 instance into the Application Server 8.*x* server1 target running under a nodeagent. However, asupgrade changes the value of the virtual server from server1 in Application Server 7 to server in Application Server 8.*x*.

These parameter values will cause Delegated Administrator to be redeployed to the instance in which Access Manager is deployed.

Index

J

L

M

N

P

S

T