Sun Java System Application Server 9.1 Update 1 Upgrade and Migration Guide

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

Part No: 820–3850–12
May 2009

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

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

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

Copyright 2009 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

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

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

090522@22510

Contents

Preface

This guide explains how to upgrade and migrate Java™ applications from the Sun Java System Application Server 8.x to the Sun Java SystemApplication Server 9.1 product line. This guide also explains how to migrate Java applications from Sun ONE Application Server 6.x/7 (also known as iPlanet Application Server), Java Enterprise Edition (Java EE™) Reference Implementation (RI) 1.3 Application Server, Sun Java System Application Server 8.x, WebLogic Application Server, WebSphere Application Server, JBoss, and so on to Application Server 9.1.

This preface contains information about and conventions for the entire Sun Java System Application Server documentation set.

Application Server Documentation Set

The Application Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for Application Server documentation is `http://docs.sun.com/coll/1343.4`. For an introduction to Application Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Application Server Documentation Set

Book Title	Description
Documentation Center	Application Server documentation topics organized by task and subject.
Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java Development Kit (JDK™), and database drivers.
Quick Start Guide	How to get started with the Application Server product.
Installation Guide	Installing the software and its components.
Deployment Planning Guide	Evaluating your system needs and enterprise to ensure that you deploy the Application Server in a manner that best suits your site. General issues and concerns that you must be aware of when deploying the server are also discussed.
Application Deployment Guide	Deployment of applications and application components to the Application Server. Includes information about deployment descriptors.

TABLE P–1 Books in the Application Server Documentation Set *(Continued)*

Book Title	Description
Developer's Guide	Creating and implementing Java Platform, Enterprise Edition (Java EE platform) applications intended to run on the Application Server that follow the open Java standards model for Java EE components and APIs. Includes information about developer tools, security, debugging, and creating lifecycle modules.
Java EE 5 Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications.
Java WSIT Tutorial	Developing web applications using the Web Service Interoperability Technologies (WSIT). Describes how, when, and why to use the WSIT technologies and the features and options that each technology supports.
Administration Guide	System administration for the Application Server, including configuration, monitoring, security, resource management, and web services management.
High Availability Administration Guide	Post-installation configuration and administration instructions for the high-availability database.
Administration Reference	Editing the Application Server configuration file, `domain.xml`.
Upgrade and Migration Guide	Upgrading from an older version of Application Server or migrating Java EE applications from competitive application servers. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
Performance Tuning Guide	Tuning the Application Server to improve performance.
Troubleshooting Guide	Solving Application Server problems.
Error Message Reference	Solving Application Server error messages.
Reference Manual	Utility commands available with the Application Server; written in man page style. Includes the `asadmin` command line interface.

Related Documentation

Application Server can be purchased by itself or as a component of Sun Java Enterprise System (Java ES), a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If you purchased Application Server as a component of Java ES, you should be familiar with the system documentation at `http://docs.sun.com/coll/1286.3`. The URL for all documentation about Java ES and its components is `http://docs.sun.com/prod/entsys.5`.

For documentation about other stand-alone Sun Java System server products, go to the following:

- Message Queue documentation (`http://docs.sun.com/coll/1343.4`)
- Directory Server documentation (`http://docs.sun.com/coll/1224.1`)
- Web Server documentation (`http://docs.sun.com/coll/1308.3`)

A Javadoc™ tool reference for packages provided with the Application Server is located at `http://glassfish.dev.java.net/nonav/javaee5/api/index.html`. Additionally, the following resources might be useful:

- The Java EE 5 Specifications (`http://java.sun.com/javaee/5/javatech.html`)
- The Java EE Blueprints (`http://java.sun.com/reference/blueprints/index.html`)

For information on creating enterprise applications in the NetBeans™ Integrated Development Environment (IDE), see `http://www.netbeans.org/kb/55/index.html`.

For information about the Java DB database included with the Application Server, see `http://developers.sun.com/javadb/`.

The GlassFish Samples project is a collection of sample applications that demonstrate a broad range of Java EE technologies. The GlassFish Samples are bundled with the Java EE Software Development Kit (SDK), and are also available from the GlassFish Samples project page at `https://glassfish-samples.dev.java.net/`.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P–2 Default Paths and File Names

Placeholder	Description	Default Value

as-install	Represents the base installation directory for Application Server.	Java ES installations on the Solaris™ operating system:
		`/opt/SUNWappserver/appserver`
		Java ES installations on the Linux operating system:
		`/opt/sun/appserver/`
		Other Solaris and Linux installations, non-root user:
		user's-home-directory`/SUNWappserver`
		Other Solaris and Linux installations, root user:
		`/opt/SUNWappserver`
		Windows, all installations:
		SystemDrive`:\Sun\AppServer`

TABLE P–2 Default Paths and File Names *(Continued)*

Placeholder	Description	Default Value

domain-root-dir	Represents the directory containing all domains.	Java ES Solaris installations:
		`/var/opt/SUNWappserver/domains/`
		Java ES Linux installations:
		`/var/opt/sun/appserver/domains/`
		All other installations:
		as-install`/domains/`

domain-dir	Represents the directory for a domain. In configuration files, you might see *domain-dir* represented as follows: `${com.sun.aas.instanceRoot}`	*domain-root-dir*`/`*domain-dir*
instance-dir	Represents the directory for a server instance.	*domain-dir*`/`*instance-dir*

Typographic Conventions

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

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	A placeholder to be replaced with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file.

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P–4 Symbol Conventions

Symbol	Description	Example	Meaning
`[ ]`	Contains optional arguments and 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.
`${ }`	Indicates a variable reference.	`${com.sun.javaRoot}`	References the value of the `com.sun.javaRoot` variable.
-	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.

Documentation, Support, and Training

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

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

Searching Sun Product Documentation

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

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

For example, to search for “broker,” type the following:

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

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

Third-Party Web Site References

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

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

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 820-3850.

◆ ◆ ◆ CHAPTER 1

Application Server Compatibility Issues

Application Server 9.1 Update 1 is binary compatible with Application Server 9.1, 8.2, 8.1, 8.0, and 7.x. Java applications that run on versions 8.1, 8.0, and 7.x also work on Application Server 9.1 except for the incompatibilities listed in this chapter.

The topics in this chapter discuss the incompatibilities in the following areas:

Application Client Interoperability

Application clients use EJBs, web services, or other enterprise components that are in the application server (on the server side). The application client and the application server must use the same version and implementation of the RMI-IIOP protocol. Application Server 9.1 Update 1 does not support communication between different versions of the protocol implementation. You cannot run application clients with one version of the application server runtime with a server that has a different version. Most often, this would happen if you upgraded the server but had not upgraded all the application client installations.

You can use the Java Web Start support to distribute and launch the application client. If the runtime on the server has changed since the end-user last used the application client, Java Web Start automatically retrieves the updated runtime. Java Web Start enables you to keep the clients and servers synchronized and using the same runtime.

HTTP File Caching

HTTP file caching, which was present in Application Server 8, has been discontinued in Application Server 9.1 Update 1.

Default Admin Port

The default admin port in Application Server 7 was 4848. The default port in Application Server 8.x was 4849. In Application Server 9.1 Update 1, the default port is 4848.

`domain.xml` Elements

If you have not configured message-level security providers for a server instance, Application Server 8 applies default configurations from the Domain Administration Server (DAS). Application Server 9.1 Update 1 does not apply default configurations. You need to manually introduce the message-level security providers — `ClientProvider` and `ServerProvider` — for each server instance that wants to use message-level security. If you have upgraded from an older version to Application Server 9.1 Update 1, the Upgrade tool does not add these missing elements in the `domain.xml` file.

Deprecated Attributes

The `anonymous-role` attribute is present in the DTD but the use of this attribute is deprecated. This attribute has been removed from the template that generates `domain.xml`. The `forced-response-type` and `default-response-type` attributes are deprecated. Use `forced-type` and `default-type` instead.

System Properties

The default security policy of Application Server 9.1 Update 1 does not allow you to change some system properties. For example, in Application Server 8, the read/write permission of `java.util.PropertyPermission` property is `"*", "read,write";`. In Application Server 9.1 Update 1 the read/write permission for `java.util.PropertyPermission` is `"*", "read";`.

Implicit URL Rewriting

Application Server 6.x supported implicit URL rewriting, in which the web connector plugin parsed the HTML stream being sent to the browser and appended session IDs to attributes such as `href=` and `frame=`. In Application Server 7,8, and Application Server 9.1 Update 1, this feature is not available. You need to review your applications and use `encodeURL` and `encodeRedirectURL` on every URL that the applications present to clients (such as mobile phones) that do not support cookies.

Web Server Features

The following web-server-specific features are no longer supported in version Application Server 9.1:

- `cgi-bin`, `shtml`
- Simple Network Management Protocol (SNMP) support
- Netscape API (NSAPI) plugin APIs
- Native-content-handling features
- Web server tools (`flexanlg`, `htpasswd`)
- HTTP QoS
- Web server configuration files (`*.conf`, `*.acl`, `mime.types`)
- Web server-specific log rotation facility
- Watch dog process (`appserv-wdog`)

Realms

The upgrade tool transfers the realms and role mapping configurations, any custom realm classes, and file-based user keyfiles for each domain. The XML tag, `security-service`, defines the realms and role mapping configuration. This tag is defined in `sun-server_1_0.dtd` and `sun-domain_1_0.dtd`. For Application Server 8, the tag data resides in the `server.xml` and for in Application Server 9.1 Update 1, in `domain.xml`.

The upgrade tool locates the class file defined for custom realms and makes it available to the Application Server 9.1 Update 1 environment. The custom realm class is defined in the class name attribute of tag `auth-realm`. In the `security-service` tag, the `default-realm` attribute points to the realm the server is using. It must point to one of the configured `auth-realm` names. The default realm is file If the class name for `default-realm` cannot be found, the upgrade tool will log this as an error.

The package names of the security realm implementations have been renamed from `com.iplanet.ias.security.auth.realm` in Application Server 8 to `com.sun.enterprise.security.auth.realm` in Application Server 9.1 Update 1. Custom realms written using the `com.iplanet.*` classes must be modified.

The `com.sun.enterprise.security.AuthenticationStatus` class has been removed.

The `com.sun.enterprise.security.auth.login.PasswordLoginModule` `authenticate` method implementation has changed as follows:

```
/**
    * Perform authentication decision.
    * <P> Note: AuthenticationStatus and AuthenticationStatusImpl
    * classes have been removed.
    * Method returns silently on success and returns a LoginException
    * on failure.
    *
    * @return void authenticate returns silently on
    *               successful authentication.
    * @throws LoginException on authentication failure.
    *
    */
abstract protected void authenticate()
    throws LoginException;
```

Sun Deployment Descriptor: `sun-web.xml`

In Application Server 8, the default value for the optional attribute delegate was false. In Application Server 9.1 Update 1, this attribute defaults to true. This change means that by default the Web application classloader first delegates to the parent classloader before attempting to load a class by itself.

The `encodeCookies` Property

URL encoding of cookies is performed, if the `encodeCookies` property of the `sun-web-app` element in the `sun-web.xml` file is set to true. In Application Server 8, the default value of the `encodeCookies` property was true. This property was not present in Application Server 8. In Application Server 9.1 Update 1, the default value is false.

URL encoding of cookies is unnecessary. Setting this property to true is strongly discouraged. This property is provided only for those rare applications that depended on this behavior in Application Server 8.

CORBA Performance Option

In Application Server 8, users were able to specify the following system property to optionally turn on some Object Request Broker (ORB) performance optimization:

```
-Djavax.rmi.CORBA.UtilClass=com.iplanet.ias.util.orbutil.IasUtilDelegate
```

The ORB performance optimization is turned on, by default, in Application Server 9.1 Update 1. If you are using the preceding system property reference, you must remove it to avoid interfering with the default optimization.

File Formats

In Application Server 9.1 Update 1, `domain.xml` is the main server configuration file. In Application Server 7, the main server configuration file was `server.xml`. The DTD file of `domain.xml` is found in `lib/dtds/sun-domain_1_1.dtd`. The upgrade tool included in Application Server 9.1 Update 1 can be used to move from `server.xml`in Application Server 8 to `domain.xml` in Application Server 9.1 Update 1.

The `lib/dtds/sun-domain_1_1.dtd` file for Application Server 9.1 Update 1 is fully backward compatible with the corresponding file for Application Server 8, `sun-domain_1_0.dtd`.

In general, the configuration file formats are *not* backward compatible. The following configuration files are *not* supported:

- `*.conf`
- `*.acl`
- `mime.types`
- `server.xml` (replaced by `domain.xml`)

Cluster Scripts

The `clsetup` and `cladmin` scripts in Application Server 8 are not supported in Application Server 9.1 Update 1. In Application Server 9.1 Update 1, the `asadmin configure-ha-cluster` command replaces the `clsetup` script, and `asadmin` commands that operate on clusters replace the commands supported by the `cladmin` script. For more information about the `asadmin` commands, see the *Sun Java System Application Server 9.1 Update 1-9.1 Update 2 Reference Manual*.

Primary Key Attribute Values

In Application Server 8, it was possible to change any field (in the Admin Console) or attribute (in the Command Line Interface (CLI)). In Application Server 9.1 Update 1, a field or attribute that is the primary key of an item cannot be changed. However, an item can be deleted and then recreated with a new primary key value. In most cases, the primary key is a name, ID, reference, or JNDI name. The following table lists the primary keys that cannot be changed.

Note – In the `domain.xml` file, a field or attribute is called an *attribute*, and an item is called an *element*. For more information about `domain.xml`, see the *Sun Java System Application Server 9.1 Administration Reference*.

TABLE 1–1 Primary Key Attributes

Item	Primary Key Field or Attribute
`admin-object-resource`	`jndi-name`
`alert-subscription`	`name`
`appclient-module`	`name`
`application-ref`	`ref`
`audit-module`	`name`
`auth-realm`	`name`
`cluster-ref`	`ref`

TABLE 1–1 Primary Key Attributes *(Continued)*

Item	Primary Key Field or Attribute
cluster	name
config	name
connector-connection-pool	name
connector-module	name
connector-resource	jndi-name
custom-resource	jndi-name
ejb-module	name
external-jndi-resource	jndi-name
http-listener	id
iiop-listener	id
j2ee-application	name
jacc-provider	name
jdbc-connection-pool	name
jdbc-resource	jndi-name
jms-host	name
jmx-connector	name
lb-config	name
lifecycle-module	name
mail-resource	jndi-name
message-security-config	auth-layer
node-agent	name
profiler	name
element-property	name
provider-config	provider-id
resource-adapter-config	resource-adapter-name
resource-ref	ref
security-map	name
server	name

TABLE 1–1 Primary Key Attributes *(Continued)*

Item	Primary Key Field or Attribute
`server-ref`	`ref`
`system-property`	`name`
`thread-pool`	`thread-pool-id`
`virtual-server`	`id`
`web-module`	`name`
`persistence-manager-factory-resource`	`jndi-name`

Command Line Interface: hadbm

The following table lists options for the command line utility `hadbm` that are no longer supported. For more information about the `hadbm` commands, see the *Sun Java System Application Server 9.1 Update 1-9.1 Update 2 Reference Manual*.

TABLE 1–2 Unsupported `hadbm` Options

Option	Unsupported in Subcommands
`--inetdsetup`	Not supported for the `addnodes` subcommand.
`--inetd`	Not supported for the `create` subcommand.
`--inetdsetupdir`	Not supported for the `create` subcommand.
`--configpath`	Not supported for the `create` subcommand.
`--set managementProtocol`	Not supported for the `create` subcommand.
`--set DataDeviceSize` `--set TotalDatadeviceSizePerNode`	Not supported for the `create` or `set` subcommand.

Command Line Interface: start-appserv and stop-appserv

The `start-appserv` and `stop-appserv` commands are deprecated. Use of these commands results in a warning. Use `asadmin start-domain` and `asadmin stop-domain` instead.

In Application Server 9.1 Update 1, the Log Messages to Standard Error field has been removed from the Admin Console. The `log-to-console` attribute in the `domain.xml` file is deprecated and ignored. The `asadmin set` command has no effect on the `log-to-console` attribute. Use the `---verbose` option of the `asadmin start-domain` command to print messages to the

window in which you executed the `asadmin start-domain` command. This option works only if you execute the `asadmin start-domain` command on the machine that has the domain you are starting.

Command Line Interface: asadmin

The following sections describe changes to the command line utility `asadmin`:

For more information about the `asadmin` commands, see the *Sun Java System Application Server 9.1 Update 1-9.1 Update 2 Reference Manual*.

asadmin Subcommands

Subcommands are backward compatible except as noted below.

The `reconfig`subcommand is deprecated and ignored.

The following subcommands are not supported in Application Server 9.1 Update 1:

- `show-instance-status` (use `list-instances`)
- `restart-instance` (use `stop-instance` followed by `start-instance`)
- `configure-session-persistence` (renamed to `configure-ha-persistence`)
- `create-session-store` (renamed to `create-ha-store`)
- `clear-session-store` (renamed to `clear-ha-store`)

The following subcommands are no longer supported in Application Server 9.1 Update 1. The software license key and web core were removed, and Application Server 9.1 Update 1 no longer supports controlled functions from web server features.

- `install-license`
- `display-license`
- `create-http-qos`
- `delete-http-qos`
- `create-mime`
- `delete-mime`
- `list-mime`
- `create-authdb`
- `delete-authdb`

- `list-authdbs`
- `create-acl`
- `delete-acl`
- `list-acls`

Error Codes for Start and Stop Subcommands

For Application Server 8, the error codes for the `start` and `stop` subcommands of the `asadmin` command were based on the desired end state. For example, for `asadmin start-domain`, if the domain was already running, the exit code was 0 (success). If domain startup failed, the exit code was 1 (error).

For Application Server 9.1 Update 1, the exit codes are based on whether the commands execute as expected. For example, the `asadmin start-domain` command returns exit code 1 if the domain is already running or if domain startup fails. Similarly, `asadmin stop-domain` returns exit code 1 if the domain is already not running or cannot be stopped.

Deprecated and Unsupported Options

Options in the following table are deprecated or no longer supported.

TABLE 1–3 Deprecated and Unsupported `asadmin` Options

Option	Deprecated or Unsupported in Subcommands
`--acceptlang`	Deprecated for the `create-virtual-server` subcommand.
`--acls`	Deprecated for the `create-virtual-server` subcommand.
`--adminpassword`	Deprecated for all relevant subcommands. Use `--passwordfile` instead.
`--blockingenabled`	Deprecated for the `create-http-listener` subcommand.
`--configfile`	Deprecated for the `create-virtual-server` subcommand.
`--defaultobj`	Deprecated for the `create-virtual-server` subcommand.
`--domain`	Deprecated for the `stop-domain` subcommand.
`--family`	Deprecated for the `create-http-listener` subcommand.
`--instance`	Deprecated for all remote subcommands. Use `--target` instead.
`--mime`	Deprecated for the `create-virtual-server` subcommand.
`--optionsfile`	No longer supported for any commands.
`--password`	Deprecated for all remote subcommands. Use `--passwordfile` instead.

TABLE 1–3 Deprecated and Unsupported `asadmin` Options *(Continued)*

Option	Deprecated or Unsupported in Subcommands
`--path`	Deprecated for the `create-domain` subcommand. Use `--domaindir` instead.
`--resourcetype`	Deprecated for all relevant subcommands. Use `--restype` instead.
`--storeurl`	No longer supported for any commands.
`--target`	Deprecated for all `jdbc-connection-pool`, `connector-connection-pool`, `connector-security-map`, and `resource-adapter-config` subcommands.
`--type`	Deprecated for all relevant subcommands.

Dotted Names

The following use of dotted names in `asadmin get` and `set` subcommands are not backward compatible:

- The default server name is `server` instead of `server1`.
- server_instance`.resource` becomes `domain.resources.resource`.
- server_instance`.app-module` becomes `domain.applications.app-module`.
- Attributes names format is different. For example,`poolResizeQuantity` is now `pool-resize-quantity`.
- Some aliases supported in Application Server 8 are not supported in Application Server 9.1 Update 1 .

In Application Server 9.1 Update 1, the `--passwordfile` option of the `asadmin` command does not read the `password.conf` file, and the upgrade tool does not upgrade this file. For information about creating a password file in Application Server 9.1 Update 1, see the *Sun Java System Application Server 9.1 Administration Guide*.

This table displays a one-to-one mapping of the incompatibilities in dotted names between Application Server 8 and 9.1. The compatible dotted names are not listed in this table.

TABLE 1–4 Incompatible Dotted Names Between Versions

Application Server 7 Dotted Names	Application Server 9.1 Dotted Names
server_instance`.http-listener.` *listener_idserver_instance*`.http-service.` `http-listener.`*listener_id*	*server_instance*`.http-service` `.http-listener.`*listener_id* *config_name*`.http-service` `.http-listener.`*listener_id*
server_instance`.orb`*server_instance*`.iiop-service`	*server_instance*`.iiop-service`*config_name* `.iiop-service`

TABLE 1–4 Incompatible Dotted Names Between Versions *(Continued)*

Application Server 7 Dotted Names	Application Server 9.1 Dotted Names
server_instance.orblistener*server_instance*.iiop-listener	*server_instance*.iiop-service.iiop-listener.*listener_id* *config_name*.iiop-service.iiop-listener.*listener_id*
server_instance.jdbc-resource.*jndi_name*	*server_instance*.resources.jdbc-resource.*jndi_name* domain.resources.jdbc-resource.*jndi_name*
server_instance.jdbc-connection-pool.*pool_id*	*server_instance*.resources.jdbc-connection-pool.*pool_id*domain.resources.jdbc-connection-pool.*pool_id*
server_instance.external-jndi-resource.*jndi_nameserver_instance*.jndi-resource.*jndi_name*	*server_instance*.resources.external-jndi-resource.*jndi_name*domain.resources.external.jndi-resource.*jndi_name*
server_instance.custom-resource.*jndi_name*	*server_instance*.resources.custom-resource.*jndi_name* domain.resources.custom-resource.*jndi_name*
server_instance.web-container.logLevel (see note below)	*server_instance*.log-service.module-log-levels.web-container*config_name*.log-service.module-log-levels.web-container
server_instance.web-container.monitoringEnabled (see note below)	*server_instance*.monitoring-service.module-monitoring-levels.web-container*config_name*.monitoring-service.module-monitoring-levels.web-container
server_instance.j2ee-application.*application_nameserver_instance*.application.*application_name*	*server_instance*.applications.j2ee-application.*application_name* domain.applications.j2ee-application.*application_name*
server_instance.ejb-module.*ejb-module_name*	*server_instance*.applications.ejb-module.*ejb-module_name*domain.applications.ejb-module.*ejb-module_name*
server_instance.web-module.*web-module_name*	*server_instance*.applications.web-module.*web-module_name*domain.applications.web-module.*web-module_name*
server_instance.connector-module.*connector_module_name*	*server_instance*.applications.connector-module.*connector_module_name* domain.applications.connector-module.*connector_module_name*

TABLE 1–4 Incompatible Dotted Names Between Versions *(Continued)*

Application Server 7 Dotted Names	Application Server 9.1 Dotted Names
server_instance.lifecycle-module.*lifecycle_module_name*	*server_instance*.applications.lifecycle-module.*lifecycle_module_name* domain.application.lifecycle-module.*lifecycle_module_name*
server_instance.virtual-server-class	N/A*
server_instance.virtual-server.*virtual-server_id*	*server_instance*.http-service.virtual-server.virtual-server_id*config_name* .http-service.virtual-server.virtual-server_id
server_instance.mime.*mime_id*	N/A*
server_instance.acl.*acl_id*	N/A*
server_instance.virtual-server.*virtual-server_id*.auth-db.*auth-db_id*	N/A*
server_instance.authrealm.*realm_idserver_instance*.security-service.authrealm.*realm_id*	*server_instance*.security-service.auth-realm.*realm_idconfig_name*.security-service-auth-realm.*realm_id*
server_instance.persistence-manager-factory-resource.*jndi_nameserver_instance*.resources.persistence-manager-factory-resource.*jndi_name*	*server_instance*.resources.persistence-manager-factory-resource.*jndi_name*domain.resources.persistence-manager-factory-resource.*jndi_name*
server_instance.http-service.acl.*acl_id*	N/A*
server_instance.mail-resource.*jndi_name*	*server_instance*.resources.mail-resource.*jndi_name*domain.resources.mail-resource.*jndi_name*
server_instance.profiler	*server_instance*.java-config.profiler*config_name*.java-config.profiler

* — These attribute names in Application Server 7 do not correspond directly with Application Server 8.2 dotted names.

Tokens in Attribute Values

The `asadmin get` command shows raw values in Application Server 9.1 Update 1 instead of resolved values as in Application Server 8. These raw values may be tokens. For example, execute the following command:

```
asadmin get domain.log-root
```

The preceding command displays the following value:

```
${com.sun.aas.instanceRoot}/logs
```

Nulls in Attribute Values

In Application Server 8, attributes with no values contained null. This caused problems in attributes that specified paths. In Application Server 9.1 Update 1, attributes with no values contain empty strings, as they did in Application Server 8.

◆ ◆ ◆ CHAPTER 2

Upgrading an Application Server Installation

The Upgrade tool, which is bundled with Application Server 9.1 Update 1, replicates the configuration of a previously installed server in the target installation. The Upgrade tool assists in upgrading the configuration, applications, and certificate data from an earlier version of the Application Server to Application Server 9.1 Update 1. To view a list of the older Application Server versions from which you can upgrade, refer Table 2–1

This chapter discusses the following topics:

Upgrade Overview

The following table shows supported Application Server upgrade paths for file-based installations.

In this table, PE indicates Platform Edition and EE indicates Enterprise Edition.

TABLE 2–1 Supported Upgrade Paths (file-based)

Source Installation	Application Server 9.1 Update 1
Application Server 7.X PE/EE/SE	Not supported

TABLE 2–1 Supported Upgrade Paths (file-based) *(Continued)*

Source Installation	Application Server 9.1 Update 1
Application Server 8.0 PE	Supported. Upgrade from 8.0 PE domain to 9.1 Update 1 developer domain is supported.
Application Server 8.1 PE	Supported. Upgrade from 8.1 PE domain to 9.1 Update 1 developer domain is supported.
Application Server 8.1 EE	Supported. Upgrade from 8.1 EE domain to 9.1 Update 1 enterprise domain is supported.
Application Server 8.2 PE	Supported. Upgrade from 8.2 PE domain to 9.1 Update 1 developer domain is supported.
Application Server 8.2 EE	Supported. Upgrade from 8.2 EE domain to 9.1 Update 1 enterprise domain is supported.
Application Server 9.0 PE	Supported. Upgrade from 9.0 PE domain to 9.1 Update 1 developer domain is supported.
Application Server 9.1 or GlassFish v2 developer domain	Supported. You can upgrade from 9.1/GlassFish v2 developer domain to 9.1 Update 1 developer domain.
Application Server 9.1 or GlassFish v2 cluster domain	Supported. You can upgrade from 9.1/GlassFish v2 cluster domain to 9.1 Update 1 cluster domain.
Application Server 9.1 enterprise domain	Supported. You can upgrade from 9.1 enterprise domain to 9.1 Update 1 enterprise domain.
GlassFish v2 Update 1 developer or cluster domain	Supported.

Note – Only the enterprise profile supports upgrades from Application Server Enterprise Edition 8.x.

The following table lists the upgrade paths for package-based installations. For more details, see "Upgrading From Java Enterprise System" on page 36.

TABLE 2–2 Supported Upgrade Paths (package-based)

Source Installation	Application Server 9.1 Update 1
Java Enterprise System 3 (Application Server 8.1)	Upgrade to Java ES 5 Update 1 first.
Java Enterprise System 4 (Application Server 8.1 Update 2)	Upgrade to Java ES 5 Update 1 first.

TABLE 2–2 Supported Upgrade Paths (package-based) *(Continued)*

Source Installation	Application Server 9.1 Update 1
Java Enterprise System 5 (Application Server 8.2)	Upgrade to Application Server 9.1 IFR, which is included as an optional download. Then, apply the appropriate patch to upgrade to Enterprise Server 2.1.
Java Enterprise System 5 Update 1 (Application Server 8.2 Update 2, Application Server 9.1 IFR as optional download)	Upgrade to Application Server 9.1 IFR, which is included as an optional download. Then, apply the appropriate patch to upgrade to Enterprise Server 2.1.

Upgrade Tool Interfaces

You can use the tool through the command-line interface (CLI) or the GUI.

To use the Upgrade tool in GUI mode, issue the `asupgrade` command with no options.

To run the Upgrade tool in CLI mode, invoke the `asupgrade` command with the `-c`/`--console` option. You can run the upgrade CLI in the interactive or non-interactive mode. If you supply all required arguments when invoking `asupgrade` on the console, the upgrade is performed in non-interactive mode and no further input is required. For a complete list of `asupgrade` options, refer Table 2–3. If you invoke the tool only with the `-c`/`--console` option, the tool enters the interactive CLI mode, where the user is asked for a series of inputs.

Note – Ensure that the `-c`/`--console` option is the first option in the command line, if you want to run `asupgrade` in CLI mode.

Upgrade Terminology

The following are important terms related to the upgrade process:

- Source Server: the installation from which you are upgrading to the new version.
- Target Server: the installation to which you are upgrading.
- Domains Root : the directory where the domains are created. This directory, by default, is the location specified as `AS_DEF_DOMAINS_PATH` in the `asenv.conf` file (on Solaris) or the `asenv.bat` file (on Windows).
- Domain Directory or *domain-dir*: the directory (within the Domains Root) corresponding to a specific domain. All the configuration and other data pertaining to the domain exists in this directory.
- Install Root: the directory where the Application Server is installed.
- Administration User Name: Name of the user who administers the server. This term refers to the admin user of the Application Server installation from which you want to upgrade.

- Password: Administration user's password to access the Domain Administration Server (DAS)(8-character minimum) of the Application Server installation from which you want to upgrade.
- Master Password: SSL certificate database password used in operations such as Domain Administration Server startup. This term refers to the master password of the Application Server installation from which you want to upgrade.

Upgrade Tool Functionality

The Upgrade Tool migrates the configuration, deployed applications, and certificate databases from an earlier version of the Application Server to the current version. The Upgrade Tool does not upgrade the binaries of the Application Server. The installer is responsible for upgrading the binaries. Database migrations or conversions are also beyond the scope of this upgrade process.

Only those instances that do not use Sun Java System Web Server-specific features are upgraded seamlessly. Configuration files related to HTTP path, CGI bin, SHTML, and NSAPI plug-ins are not be upgraded.

Note – Before starting the upgrade process, make sure that you stop all server instances, node agents, and domains (in that order) in the source server (the server from which you are upgrading) and the target server (the server to which you are upgrading).

Migration of Deployed Applications

Application archives (EAR files) and component archives (JAR, WAR, and RAR files) that are deployed in the Application Server 8.x environment do not require any modification to run on Application Server 9.1 Update 1.

Applications and components that are deployed in the source server are deployed on the target server during the upgrade. Applications that do not deploy successfully on the target server must deployed manually on the target server by the user1

If a domain contains information about a deployed application and the installed application components do not agree with the configuration information, the configuration is migrated as is without any attempt to reconfigure the incorrect configurations.

Upgrade of Clusters

In Application Server 8.x, the clusters are defined in the `domain.xml` file and there is no need to specify clusters separately. Another notable difference is that in Application Server 8.x, all the instances within a cluster reside within the same domain and therefore, in the same `domain.xml` file.

Transfer of Certificates and Realm Files

The Upgrade tool transfers certificates from the source certificate database to the target. The tool transfers security policies, password files from standard, file-based realms, and custom realm classes.

Upgrade Verification

An upgrade log records the upgrade activity. The upgrade log file is named as the `upgrade.log` and is created in the domains root where the upgrade is carried out.

After you have upgrade a domain, you can see a file whose name is in the following format: `upgradedTo`<*releasenumber*>. For example, a domain that has been upgraded to 9.1 Update 1 will have a file called `upgradeTo91` in its `config` folder.

Upgrade Rollback

If an upgrade in progress is cancelled, the configuration before the upgrade was started is restored.

Note – You can cancel the upgrade process only if you are running the Upgrade Tool in GUI mode.

Upgrade Scenarios

The upgrade scenarios for application server are as follows:

- "Side-by-Side Upgrade" on page 29
- "In-Place Upgrade" on page 30

Side-by-Side Upgrade

The source server and the target server are installed on the same machine , but under different install locations. You can choose to perform this type of upgrade if you wish to have the configuration corresponding to these installations on the same machine in different locations.

An in-place upgrade involves the following sequence of tasks

1. "Upgrading Application Server Runtime Binaries" on page 30
2. "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32.
3. "To Upgrade Remote Node Agents (Side-by-side)" on page 35 (only if you have remote agents in the source application server).

If all existing node agents (from the source application server installation) run on a single machine, the upgrade tool automatically detects node agents, if any, on the source installation.

In-Place Upgrade

The target server is installed in the same installation location as the source server. You can choose to perform this type of upgrade if you wish to install the configuration (that is, the domains) in the same location as before. In this scenario, you install the binaries in the same location as the existing binaries using the installer.

An in-place upgrade involves the following sequence of tasks:

1. "Upgrading Application Server Runtime Binaries" on page 30
2. "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32.
3. "To Upgrade Remote Node Agents (In-place)" on page 34 (only if you have remote agents in the source application server).

   If all existing node agents (from the source application server installation) run on a single machine, the upgrade tool automatically detects node agents, if any, on the source installation.

Upgrading Application Server Runtime Binaries

The Upgrade tool does not update the runtime binaries of the server. The Upgrade tool upgrades the configuration information and deployed applications of a previously installed server. Before you run the Upgrade tool to upgrade the configuration data, you need to upgrade the runtime binaries. You can upgrade the binaries in one of the following ways:

- "Upgrading the Runtime Binaries for a File-Based Installation of Application Server" on page 30
- "Upgrading the Runtime Binaries for a Package-Based Installation of Application Server" on page 31

Upgrading the Runtime Binaries for a File-Based Installation of Application Server

Use the Application Server file-based installer to install the target application server binaries. You can install in-place (in the same location as your earlier installation) or side-by-side (in a different location). For instructions on how to use the file-based installer, see *Sun Java System Application Server 9.1 Update 1 Installation Guide*.

After installing the target application server binaries, run the `asupgrade` command for "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32.

Upgrading the Runtime Binaries for a Package-Based Installation of Application Server

If you had performed a package-based installation of Application Server 9.1, upgrade to Application Server 9.1 Update 1 by applying the appropriate patch. Patches for the Solaris operating system and the Linux operating system are available from the SunSolve℠ program site (`http://sunsolve.sun.com`).

▼ To Upgrade a Package-Based Installation of Application Server 9.1

This procedure applies *only* to upgrades from Application Server 9.1 to Application Server 9.1 Update 1.

1. **Stop all instances, node agents, clusters, and domains in the source Application Server.**
2. **Download the appropriate patch from the SunSolve program site (`http://sunsolve.sun.com`).**
3. **On the machine where the Application Server is installed, log in as root or become superuser.**
4. **Apply the patch.**

   `patchadd` *patch-id*

   patch-id is the patch. For example, `/var/sadm/spool/patch/128640-01`
5. **Start the Application Server.**

Next Steps Run the `asupgrade` command for "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32.

Then, log in to the Admin Console. You can register your installation of application server with Sun Connection (`http://www.sun.com/service/sunconnection/index.jsp`) from the Admin Console by clicking Common Tasks -> Registration. For detailed instructions for registering the Application Server with Sun Connection, see the Admin Console online help.

Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases

You cannot perform an upgrade if the source and target server file systems, specifically the domain root file system, are not accessible from the same machine. To perform the upgrade, the user who runs the upgrade needs to have Read permissions for the source and target directories and Write permission for the target directory.

Note – Ensure that you have stopped all instances, node agents, clusters, and domains in the source Application Server before you start the upgrade process.

For upgrading your Application Server installation you can choose:

- "To Upgrade From the Command Line" on page 32
- "To Upgrade by Using the Upgrade Tool Wizard" on page 33

To Upgrade From the Command Line

To run Upgrade Tool in command-line mode, use the `-c` option. You can run the upgrade tool in command-line mode using the following syntax:

```
asupgrade
[--console ]
[--version ]
[--help ]
[--source applicationserver_installation_domaindirectory]
[--target applicationserver_9.1u1_installation]
[--adminuser applicationserver_9.1u1_admin_user_name]
[--passwordfile passwords.txt]
```

The following table describes the command options in greater detail, including the short form, the long form, and a description.

TABLE 2–3 asupgrade Utility Command Options

Short Form	Long Form	Description
-c	--console	Launches the upgrade command line utility.
-v or -V	--version	The version of the Upgrade Tool.
-h	--help	Displays the arguments for launching the upgrade utility.
-t	--target	The domains directory of the Application Server 9.1 Update 1 installation.

TABLE 2–3 asupgrade Utility Command Options *(Continued)*

Short Form	Long Form	Description
-s	--source	The installation directory of the older Application Server installation.
-a	--adminuser	The admin user for the source server.
-f	--passwordfile	The file containing the admin password and the master password.

The following examples show how to use the `asupgrade` command-line utility to upgrade an existing application server installation to Application Server 9.1 Update 1.

This example shows how to perform a side-by-side upgrade of a Sun Java SystemApplication Server 8.x installation to Application Server 9.1 Update 1.

```
asupgrade -c --source /home/sunas8.2/domains/domain1 --target /home/sjsas9.1/domains
```

▼ To Upgrade by Using the Upgrade Tool Wizard

To start the wizard,

- On UNIX, change to the `<install_dir>/bin` directory and type `asupgrade`.

- On Windows, double click the `asupgrade` icon in the `<install_dir>/bin` directory.

If the Upgrade checkbox was selected during the Application Server installation process, the Upgrade Wizard screen automatically displays after the installation completes.

1. **In the Source Installation Directory field, enter the location of the existing installation from which to import the configuration. Enter the domain directory.**

   For example, <install-root>/domains/domain1

2. **In the Target Installation Directory field, enter the location of the Application Server installation to which to transfer the configuration. Provide the domains root directory of the target Application Server installation as the input to this field.**

3. **Provide the admin user name, the admin password, and master password of the source application server. The target domain is created with these credentials.**

4. **The Upgrade Results panel is displayed showing the status of the upgrade operation.**

5. **Click the Finish button to close the Upgrade Tool when the upgrade process is complete.**

Next Steps To upgrade node agents, see "Upgrading Node Agents" on page 34. After you complete the upgrade, start the Application Server using the `asadmin start-domain` command. Log on to the Admin Console. You can register your installation of application server from the Admin Console by clicking Common Tasks -> Registration. For step-by-step instructions on the registration process, click the Help button on the Admin Console.

Upgrading Node Agents

The upgrade tool automatically detects clusters, if any, on the source installation.

If all existing node agents (from the source Application Server installation) run on a single machine, the upgrade tool automatically detects node agents, if any, on the source installation. The user need not take any special action. If you have remote node agents running on other machines, use the following steps to perform the upgrade.

- "To Upgrade Remote Node Agents (In-place)" on page 34
- "To Upgrade Remote Node Agents (Side-by-side)" on page 35

▼ To Upgrade Remote Node Agents (In-place)

1. **Perform the upgrade to Application Server 9.1 Update 1 on `Machine A` by "Upgrading Application Server Runtime Binaries" on page 30 and "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32.**

2. **Install Application Server 9.1 Update 1 on `Machine B` without the DAS but with the Node Agent feature.**

   Note – `Machine A` is the primary machine. It runs the DAS. `Machine B` is a secondary machine, which is not running the DAS. `Machine B` runs remote node agents that are configured to communicate with `Machine A`.

   ---

3. **On `Machine A`, start each node agent using the `start-node-agent` command with the `--syncinstances` option. This option resynchronizes all associated instances. Example:**
   ```
   asadmin start-node-agent --user admin --syncinstances nodeagent1
   ```

4. **On `Machine B`, start each node agent using the `start-node-agent` command with the `--syncinstances` option. This option resynchronizes all associated instances**

▼ To Upgrade Remote Node Agents (Side-by-side)

1 **Perform the upgrade to Application Server 9.1 Update 1 on** `Machine A` **by** **"Upgrading Application Server Runtime Binaries" on page 30** **and "Upgrading Application Server Configuration, Deployed Applications, and Certificate Databases" on page 32****.**

2 **Install Application Server 9.1 Update 1 on** `Machine B` **without the DAS but with the Node Agent feature.**

Note – `Machine A` is the primary machine. It runs the DAS. `Machine B` is a secondary machine, which is not running the DAS. `Machine B` runs remote node agents that are configured to communicate with `Machine A`.

---

3 **Check the value of the** `agent.das.port` **property in the** `das.properties` **file before starting the node agent for the first time. Perform this check on the** `das.properties` **file on** `Machine B`**. The value of the** `agent.das.port` **property must reflect the same value as the** `jmx-connector` **port defined in the** `domain.xml` **file on** `Machine A`**.**

There are two ways to determine this port number:

- It is displayed as part of the message for the `start-domain(1)` command run on the DAS machine. Look for the following line in the command output of `asadmin start-domain`: `[service:jmx:rmi:///jndi/rmi://comet:8686/jmxrmi] for domain management purposes.` This line indicates that the JMX port number is 8686.
- It is recorded in the domain's `domain.xml` file on the `Machine A`. Find the `admin-service` element with `das-and-server` attribute type. Look at the `jmx-connector` sub-element for this element and find the attribute port. The value of this attribute is the port number of the JMX port.

Edit the `agent.das.port` property in the`das.properties` file on `Machine B`, as required.

4 **Check the value of the** `agent.bind.status` **property in the** `nodeagent.properties` **file before starting the node agent for the first time. Perform this check on the** `nodeagent.properties` **file on** `Machine B`**. The value of the** `agent.bind.status` **property must** `BOUND`**. Edit the** `agent.bind.status` **property in the** `nodeagent.properties` **files on** `Machine B`**, as required.**

5 **On** `Machine A`**, start each node agent using the** `start-node-agent` **command.**

6 **On** `Machine B`**, start each node agent using the** `start-node-agent` **command.**

7 **If you are using non-default admin ports, you must additionally perform the following steps:**

Before you perform these steps, ensure that you start the node agents so that the required configuration files are created. However, the startup of node agents will not be successful.

a. **Change directory to the node agents config directory (***as-install*`/nodeagents/`*nodeagent-name*`/agent/config`**). Edit the** `das.properties` **file. The value of** `agent.das.port` **property is the admin port number of the DAS machine with which this node-agent communicates. Edit this value to match the value of the jmxrmi port number of the DAS machine.**

There are two ways to find the jmxrmi port number:

- The DAS' `start-domain` command outout displays the jmxrmi value. Look for the following line: `Standard JMX Clients (like JConsole) can connect to JMXServiceURL: [service:jmx:rmi:///jndi/rmi://comet:8696/jmxrmi] for domain management purposes.` The jmxrmi port is 8696. Set `agent.das.port` to this value.
- Search the `domain.xml` file in the appropriate domain on the DAS machine. Find element `admin-service` element with `das-and-server` attribute type. Look at the `jmx-connector` sub-element for this element and find the attribute port. The value of this attribute is the port number of the jmxrmi port. Update the property value, save, and exit the file

b. **Edit the** `nodeagent.properties` **file. If the** `agent.bind.status` **property is set to** `UNBOUND`**, change the value to** `BOUND`**. Save and exit the file.**

8 **Start the node agent. The** `--syncinstances=true` **option need not be used.**

Upgrading From Java Enterprise System

▼ To Upgrade From Java Enterprise System

You can upgrade from Java Enterprise System (Java ES) 3, 4, or 5 to Application Server 9.1.

1 **Upgrade to Java ES 5 Update 1.**

For more details, see the Java ES 5 documentation on docs.sun.com. If you have already upgraded Message Queue to a version later than 4.1, upgrading to Java ES 5 Update 1 will cause Message Queue to be downgraded to version 4.1.

2 **Install Sun Java System Application Server 9.1 IFR bundle.**

This bundle is available with Java ES 5 Update 1 as an optional download.

Note – If you have already installed a JDK version that is newer than 1.5.0_12, do not choose the Install Java 2 SDK (5.0) option during the Application Server 9.1 installation. Select the Reuse Existing Java 2 SDK option instead.

3 **Use appropriate patches from SunSolve to upgrade from to Application Server 9.1.**

See "To Upgrade a Package-Based Installation of Application Server 9.1" on page 31 for patch numbers and more details.

Correcting Potential Upgrade Problems

This section addresses the following issues that could occur during an upgrade to Application Server 9.1 Update 1:

Cluster Profile Security Setting

When you upgrade a cluster domain from Application Server 9.1/GlassFish v2 to Application Server 9.1 Update 1, you could encounter problems because the security setting is incorrect for the `admin-service` whose type attribute is `das-and server` in the target `domain.xml`. The solution is to edit the `domain.xml` file in the corresponding upgraded domain and correct the setting of the `security-enabled` attribute. Look for the following statements in the `domain.xml` file.

```
<admin-service system-jmx-connector-name="system" type="das-and-server">
<|-- The JSR 160 "system-jmx-connector"-->
<jmx-connector accept-all="false" address="0.0.0.0"
auth-realm-name="admin-realm" enabled=true" name="system" port="8686"
protocol="rmi_jrmp" security-enabled="true">
```

Cluster Profile Upgrade on Windows

On Windows, when you upgrade cluster profile domains, you could encounter the following error:

```
Fatal error while backing up the domain directory
```

To resolve this error, look for and remove any hidden files in the source domain's directory and run Upgrade tool.

`_TimerPool` Resource

The datasource class used for a `jdbc-connection-pool` resource named `__TimerPool` has changed from `org.apache.derby.jdbc.EmbeddedXADataSource` in Application Server 8.x EE to `org.apache.derby.jdbc.ClientDataSource` in Application Server 9.1 Update 1. This change requires a addition of two property elements, User and Password to the `jdbc-connection-pool` element in the `domain.xml` file. Edit the Application Server 9.1 Update 1 `domain.xml` file and add the appropriate user name and password. Example:

```
<property name="User" value="APP"/> <property name="Password" value="APP"/>
```

Missing Client JAR Files

You have deployed applications that use client JARs in Application Server 8.x. You upgrade your existing installation to Application Server 9.1 Update 1. You could run into problems while trying to run these applications (that were deployed in Application Server 8.x) in Application Server 9.1 Update 1.

To solve this problem, perform the following steps:

1. After upgrade, start Application Server 9.1 Update 1.
2. Use the `asadmin get-client-stubs` command to transfer the missing client stubs to a local directory. See `get-client-stubs(1)`.
3. Run the appclient pointing to the client JAR files in the local directory.

Applications that Use JavaDB

You have deployed applications that use JavaDB databases in Application Server 8.x. You upgrade your existing installation to Application Server 9.1 Update 1. You run the `asadmin start-database` command and successfully start JavaDB. In this scenario, you could run into problems while trying to run these applications (that were deployed in Application Server 8.x) in Application Server 9.1 Update 1 because the instance directory of JavaDB in Application Server 9.1 Update 1 has changed.

To solve this problem, perform the following steps:

1. After upgrade, start Application Server 9.1 Update 1.
2. Use the `asadmin start-database` command with `--dbhome` option pointing to older (Application Server 8.x) version of JavaDB. Example `asadmin start-database --dbhome /home/johnsmith/appserver8.2/databases`
3. Deploy the migrated applications.

JVM Options That are Not Transferred

When you upgrade from a previous version of the application server, transfer of the previous configuration is required. Since the target configuration files may have new parameters and new preconfigured features, copying the old configuration files to the new server installation is not possible. The values of the old configurations must be transferred to the Application Server 9.1 Update 1 configuration format.

The following JVM options are not transferred from the source to the target installation:

- `Dorg.xml.sax.driver`
- `Dcom.sun.jdo.api.persistence.model.multipleClassLoaders`
- `Djava.util.logging.manager`
- `Dcom.sun.aas.imqLib`
- `Dcom.sun.aas.imqBin`
- `Dcom.sun.aas.webServicesLib`
- `Dcom.sun.aas.configRoot 8. Xmx<...>m`

The options that are not transferred are listed down in the upgrade log. The user can manually change such attributes in the configuration file.

Port Conflicts

After upgrading the source server to Application Server 9.1 Update 1, start the domain and then the node agent, which, by default, starts the server instances. If you have upgraded from Application Server 8.x EE, you might face problems while attempting to start the node agent. The domain, clusters, and instances have admin port set to 4849 and the node agent points to 4848. You need to manually modify the admin port to which the node agent points. To change the node agent port, edit the `agent.das.port` property in the *install_dir*`/nodeagents/`*node-agent-name*`/`*server_name*`/config/das.properties` file.

Start the Admin Console and verify that these servers are started. If any of the servers are not running, in the *install_dir*`/nodeagents/`*node-agent-name*`/`*server_name*`/logs/server.log` file, check for failures that are caused by port conflicts. If there any failures due to port conflicts, use the Admin Console and modify the port numbers so there are no more conflicts. Stop and restart the node agent and servers.

Note – The default ports in Application Server 9.1 Update 1 are:

- 4848 for admin port
- 8080 for HTTP Instance (DAS instance)
- 7676 for JMS
- 3700 for IIOP
- 8181 for HTTP_SSL.
- 3820 for IIOP_SSL
- 3920 for IIOP_MUTUALAUTH
- 8686 for JMX_ADMIN

Single Domain with Multiple Certificate Database Passwords

If the upgrade includes certificates, provide the passwords for the source PKCS12 file and the target JKS keyfile for each domain that contains certificates to be migrated. Since Application Server 8uses a different certificate store format (NSS) than that of Application Server 8 PE (JSSE), the migration keys and certificates are converted to the new format. Only one certificate database password per domain is supported. If multiple certificate database passwords are used in a single domain, make all of the passwords the same before starting the upgrade. Reset the passwords after the upgrade has been completed.

Load balancer Plug-in Problems During Side-by-Side Upgrade

While upgrading from Application Server 8.x EE to Application Server 9.1 Update 1, during a side-by-side upgrade, you will not be able to point your new 9.1 Update 1 load balancer plug-in to the old 8.x web server installation, if the load balancer plug-in is colocated with other Application server components on a single system. You need to install web server again and point the 9.1 load balancer plug-in installation to the instance belonging to the new installation.

▼ Additional HTTP Listeners

If additional HTTP listeners have been defined in the source server, those listeners need to be added to the target server after the upgrade:

1. **Start the Admin Console.**
2. **Expand Configuration.**

3. **Expand HTTP Service.**
4. **Expand Virtual Servers.**
5. **Select <server>.**
6. **In the right hand pane, add the additional HTTP listener name to the HTTP Listeners field.**
7. **Click Save when done.**

Additional HTTP and IIOP Listeners

If additional HTTP listeners or IIOP listeners have been defined in the source server, the IIOP ports must be manually updated for the target EE servers before any clustered instances are started. For example, `MyHttpListener` was defined as an additional HTTP listener in `server1`, which is part of the cluster. The other instances in the cluster also have the same HTTP listener, because server instances are symmetrical in a cluster. In the target configuration named <*cluster_name*>`-config`, this listener must be added with its port set to a system property, `{myHttpListener_HTTP_LISTENER_PORT}`. In the target server, each server instance in this cluster that uses this configuration would have system property named `myHttpListener_HTTP_LISTENER_PORT`. The value of this property for all server instances is set to the port value in the source server, `server1`. These system properties for these server instances must be manually updated with nonconflicting port numbers before the server is started.

If additional HTTP listeners have been defined in the source server, those listeners need to be added to the target server after the upgrade:

1. **Start the Admin Console.**
2. **Expand Configuration and select the appropriate** <*server*>`-config` **configuration.**
3. **Expand HTTP Service.**
4. **Expand Virtual Servers.**
5. **Select** <*server*>**.**
6. **In the right hand pane, add the additional HTTP listener name(s) to the HTTP Listeners field.**
7. **Click Save when done.**

Index