Sun GlassFish Communications Server Administration Reference

Beta

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

Part No: 820–4284
March 2008

080715@20490

Contents

Preface

This *Administration Reference* provides information about the Sun Java System Communications Application Server configuration file, `domain.xml`. This file contains most of the Communications Application Server configuration.

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

Communications Application Server Documentation Set

The Communications Communications Application Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for Communications Application Server documentation is `http://docs.sun.com/coll/1343.8`. For an introduction to Communications Application Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Communications Application Server Documentation Set

Book Title	Description
Documentation Center	Communications 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 Communications 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 Communications 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 Communications Application Server. Includes information about deployment descriptors.

TABLE P–1 Books in the Communications 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 Communications 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 Communications 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 Communications Application Server configuration file, `domain.xml`.
Upgrade and Migration Guide	Upgrading from an older version of Communications 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 Communications Application Server to improve performance.
Troubleshooting Guide	Solving Communications Application Server problems.
Error Message Reference	Solving Communications Application Server error messages.
Reference Manual	Utility commands available with the Communications Application Server; written in man page style. Includes the `asadmin` command line interface.

Related Documentation

Communications 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 Communications 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 Communications 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 Communications 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 Communications 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 820-4284.

◆ ◆ ◆ CHAPTER 1

The domain.xml File

This chapter describes the `domain.xml` configuration file in these sections:

Note – Subelements must be defined in the order in which they are listed under each **Subelements** heading in this chapter unless otherwise noted.

About the domain.xml File

The `domain.xml` file contains most of the Sun Java™ System Communications Application Server configuration. The encoding is UTF-8 to maintain compatibility with regular UNIX text editors. The `domain.xml` file is located in the domain configuration directory, which is typically *domain-dir*`/config`. This file is further described in the following sections:

Note – Settings in the Communications Application Server deployment descriptors override corresponding settings in the `domain.xml` file unless otherwise stated. For more information about the Communications Application Server deployment descriptors, see the *Sun Java System Application Server 9.1 Application Deployment Guide*.

The sun-domain_1_4.dtd File

The `sun-domain_1_4.dtd` file defines the structure of the `domain.xml` file, including the elements it can contain and the subelements and attributes these elements can have. The `sun-domain_1_4.dtd` file is located in the *as-install*`/lib/dtds` directory.

Note – Do not edit the `sun-domain_1_4.dtd` file; its contents change only with new versions of the Communications Application Server.

The `sun-domain_1_4.dtd` interface is unstable. An unstable interface might be experimental or transitional, and hence might change incompatibly, be removed, or be replaced by a more stable interface in the next release.

Elements or attributes that appear in the `sun-domain_1_4.dtd` file but are not described in this chapter are not implemented and should not be used.

For general information about DTD files and XML, see the XML specification (`http://www.w3.org/TR/REC-xml`).

Default Values

In this manual, the term *default* is used in its broader sense, and not in the specific way it is used in the XML 1.0 standard. A default value is an initial value or the value used if no value is present in the XML file. A default value can be any of the following:

- A value supplied by the XML parser when no value is found in the `domain.xml` file. The relevant element or attribute is optional.
- A value supplied by the Communications Application Server when no value is found in the `domain.xml` file and the XML parser doesn't provide a value. The relevant element or attribute is optional.
- An initial value supplied when the `domain.xml` file is created. The relevant element or attribute might or might not be optional.

When a required attribute or property has a default, this default is supplied when the `domain.xml` file is created.

Variables

Variables and variable references are needed for two reasons:

- Parts of the Communications Application Server share much configuration information but differ in specific details. For example, server instances in a cluster typically share the same configuration except for their port numbers.

- Parts of the configuration come from the system environment but must still be captured in the configuration.

Variable references appear in the `domain.xml` file as strings that begin with the characters `${` and end with the character `}`. For example, the string `${com.sun.enterprise.myVar}` is a reference to the variable `com.sun.enterprise.myVar`.

Variables are defined both outside of and within `domain.xml`. Predefined variables that exist outside of `domain.xml` are defined as Java System Properties. Within `domain.xml`, a variable is defined using the "system-property" on page 154 element or the "jvm-options" on page 100 element.

The `system-property` element's `name` attribute is the name of a variable; its `value` attribute is the definition of the variable. For example, the following `system-property` element defines a `port-number` variable with the value `6500`:

```
<system-property name="port-number" value="6500"/>
```

Multiple `system-property` subelements are permitted within "server" on page 137, "cluster" on page 36, "config" on page 38, and "domain" on page 56 elements.

A variable defined in the `jvm-options` element is a Java System Property with the `-D` flag. For example, the following `jvm-options` element defines a `port-number` variable with the value `5500`:

```
<jvm-option>-Dport-number=5500</jvm-option>
```

Multiple definitions for the same variable are permitted. The Communications Application Server determines the actual value of a variable by searching for its first definition in a strict hierarchy of the elements within `domain.xml`. The hierarchy is as follows:

```
server → cluster → config → jvm-options → domain → System
```

Implicit in this hierarchy is the notion of reference and containment. A variable referenced in a `server` element is only looked up:

- In the `cluster` element referenced by that specific `server`
- In the `config` element that references that specific `server`
- In the `jvm-options` subelements of the `config` element referenced by that `server`

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Element Referencing

One element *references* another when an attribute of the referencing element has the same value as an attribute of the referenced element. For example, the "application-ref" on page 28 element references an application or module that is deployed to its parent "server" on page 137 element. The `application-ref` element's `ref` attribute has the same value as the `name` attribute of a "lifecycle-module" on page 103, "j2ee-application" on page 86, "ejb-module" on page 61, "web-module" on page 171, "connector-module" on page 45, or "appclient-module" on page 27 element.

The referencing `application-ref` element might look like this:

```
<application-ref ref="MyServlet"/>
```

The referenced `web-module` element might look like this:

```
<web-module name="MyServlet" location="myservletdir"/>
```

Element Hierarchy

The element hierarchy for the `domain.xml` file is as follows. To make the hierarchy more readable, elements having "property" on page 125 as their last or only subelement are marked with a `P`, and the `property` subelements are not shown. Parent/child relationships between elements are shown, but not cardinality. For those details, see the element descriptions.

```
domain      P
.    applications
.    .    lifecycle-module      P
.    .    .    description
.    .    j2ee-application      P
.    .    .    description
.    .    .    web-service-endpoint
.    .    .    .    registry-location
.    .    .    .    transformation-rule
.    .    web-module      P
.    .    .    description
.    .    .    web-service-endpoint
.    .    .    .    registry-location
.    .    .    .    transformation-rule
.    .    ejb-module      P
.    .    .    description
.    .    .    web-service-endpoint
.    .    .    .    registry-location
.    .    .    .    transformation-rule
.    .    connector-module      P
.    .    .    description
```

```
.    .    appclient-module     P
.    .    .    description
.    .    mbean     P
.    .    .    description
.    .    extension-module     P
.    .    .    description
.    resources
.    .    custom-resource     P
.    .    .    description
.    .    external-jndi-resource     P
.    .    .    description
.    .    jdbc-resource     P
.    .    .    description
.    .    mail-resource     P
.    .    .    description
.    .    persistence-manager-factory-resource     P
.    .    .    description
.    .    admin-object-resource     P
.    .    .    description
.    .    connector-resource     P
.    .    .    description
.    .    resource-adapter-config     P
.    .    jdbc-connection-pool     P
.    .    .    description
.    .    connector-connection-pool     P
.    .    .    description
.    .    .    security-map
.    .    .    .    principal
.    .    .    .    user-group
.    .    .    .    backend-principal
.    configs
.    .    config     P
.    .    .    sip-service     P
.    .    .    .    access-log
.    .    .    .    sip-listener     P
.    .    .    .    .    ssl
.    .    .    .    request-processing
.    .    .    .    keep-alive
.    .    .    .    connection-pool
.    .    .    .    sip-protocol     P
.    .    .    .    .    sip-link
.    .    .    .    .    sip-timers
.    .    .    http-service     P
.    .    .    .    access-log
.    .    .    .    request-processing
.    .    .    .    keep-alive
.    .    .    .    connection-pool
.    .    .    .    http-protocol
```

```
. . . . http-listener P
. . . . . ssl
. . . . virtual-server P
. . . . . http-access-log
. . . iiop-service
. . . . orb P
. . . . ssl-client-config
. . . . . ssl
. . . . iiop-listener P
. . . . . ssl
. . . admin-service P
. . . . jmx-connector P
. . . . . ssl
. . . . das-config P
. . . connector-service
. . . web-container P
. . . . session-config
. . . . . session-manager
. . . . . . manager-properties P
. . . . . . store-properties P
. . . . . session-properties P
. . . ejb-container P
. . . . ejb-timer-service P
. . . mdb-container P
. . . sip-container P
. . . . session-config
. . . . . session-manager
. . . . . . manager-properties P
. . . . . . store-properties P
. . . . . session-properties P
. . . . stack-config P
. . . . . stack-layer P
. . . jms-service P
. . . . jms-host P
. . . log-service P
. . . . module-log-levels P
. . . security-service P
. . . . auth-realm P
. . . . jacc-provider P
. . . . audit-module P
. . . . message-security-config
. . . . . provider-config P
. . . . . . request-policy
. . . . . . response-policy
. . . . identity-assertion-trust
. . . . . trusted-entity
. . . . . . ip-address
. . . . . . host-name
```

```
. . . . . principal
. . . . . trust-handler P
. . . transaction-service P
. . . monitoring-service P
. . . . module-monitoring-levels P
. . . diagnostic-service P
. . . java-config P
. . . . profiler P
. . . . . jvm-options
. . . . jvm-options
. . . availability-service P
. . . . web-container-availability P
. . . . ejb-container-availability P
. . . . sip-container-availability P
. . . . jms-availability P
. . . . converged-load-balancer
. . . . . proxy P
. . . thread-pools
. . . . thread-pool
. . . alert-service P
. . . . alert-subscription
. . . . . listener-config P
. . . . . filter-config P
. . . group-management-service P
. . . management-rules
. . . . management-rule P
. . . . . description
. . . . . event P
. . . . . description
. . . system-property
. . . . description
. servers
. . server P
. . . application-ref
. . . resource-ref
. . . system-property
. . . . description
. clusters . . cluster P
. . . server-ref
. . . . health-checker
. . . resource-ref
. . . application-ref
. . . system-property
. . . . description
. node-agents
. . node-agent P
. . . jmx-connector P
. . . . ssl
```

```
.    .    .    auth-realm      P
.    .    .    log-service      P
.    .    .    .    module-log-levels      P
.    lb-configs
.    .    lb-config      P
.    .    .    cluster-ref
.    .    .    .    health-checker
.    .    .    server-ref
.    .    .    .    health-checker
.    load-balancers
.    .    load-balancer      P
.    converged-lb-configs
.    .    converged-lb-config      P
.    .    .    converged-lb-policy      P
.    .    .    converged-lb-cluster-ref
.    .    .    server-ref
.    .    .    .    health-checker
.    system-property
.    .    description
```

Alphabetical List of Elements

A

access-log

Defines access log settings for "sip-service" on page 149 and for each "http-access-log" on page 73 subelement of each "virtual-server" on page 162.

Superelements

"http-service" on page 79, "sip-service" on page 149

Subelements

none

Attributes

The following table describes attributes for the `access-log` element.

TABLE 1–1 `access-log` Attributes

Attribute	Default	Description
`format`	`%client.name% %auth-user-name% %datetime% %request% %status% %response.length%`	(optional) Specifies the format of the access log. For a complete list of token values you can use in the format, see the online help for the Access Log tab of the HTTP Service or Session Initiation Protocol (SIP) Service page in the Admin Console.
`rotation-policy`	`time`	(optional) Specifies the condition that triggers log rotation. The only legal value is `time`, which rotates log files at the `rotation-interval-in-minutes` interval.
`rotation-interval -in-minutes`	`15` (developer profile) `1440` (cluster and enterprise profiles)	(optional) Specifies the time interval between log rotations if `rotation-policy` is set to `time`.
`rotation-suffix`	`yyyy-MM-dd` (developer profile) `yyyyMMdd-HH'h'mm'm'ss's'` (cluster and enterprise profiles)	(optional) Specifies the format of the timestamp appended to the access log name when log rotation occurs. For supported formats, see http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html. The following value is supported for backward compatibility. It results in the same format as the default. `%YYYY;%MM;%DD;-%hh;h%mm;m%ss;s`
`rotation-enabled`	`true`	(optional) If `true`, enables log rotation.

action

Specifies the action of a management rule. The action is implemented as an MBean.

Superelements

"management-rule" on page 110

Subelements

none

Attributes

The following table describes attributes for the `action` element.

TABLE 1–2 `action` Attributes

Attribute	Default	Description
`action-mbean-name`	none	Specifies the `name` of the "mbean" on page 113 that performs the action of a management rule. This MBean must implement `javax.management.NotificationListener`.

admin-object-resource

Defines an administered object for an inbound resource adapter.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `admin-object-resource` element.

TABLE 1–3 `admin-object-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `admin-object-resource` element.

TABLE 1–4 `admin-object-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`res-type`	none	Specifies the fully qualified type of the resource.
`res-adapter`	none	Specifies the name of the inbound resource adapter, as specified in the `name` attribute of a "connector-module" on page 45 element.

TABLE 1–4 `admin-object-resource` Attributes *(Continued)*

Attribute	Default	Description
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

Properties

Properties of the `admin-object-resource` element are the names of setter methods of the `adminobject-class` specified in the `adminobject` element of the `ra.xml` file. Some of the property names can be specified in the `adminobject` element itself. For example, in `jmsra`, the resource adapter used to communicate with the Sun Java System Message Queue software, `jmsra`, `Name` and `Description` are valid properties.

For a complete list of the available properties (called *administered object attributes* in the Message Queue software), see the *Sun Java System Message Queue 3.7 UR1 Administration Guide*.

admin-service

Determines whether the server instance is a regular instance, a domain administration server, or a combination.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `admin-service` element.

TABLE 1–5 `admin-service` Subelements

Element	Required	Description
"jmx-connector" on page 98	zero or more	Configures a JSR 160/255 compliant remote JMX connector.

TABLE 1–5 `admin-service` Subelements *(Continued)*

Element	Required	Description
"das-config" on page 53	only one (developer profile) zero or one (cluster and enterprise profiles)	Defines a domain administration server configuration.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `admin-service` element.

TABLE 1–6 `admin-service` Attributes

Attribute	Default	Description
`type`	`das-and-server` (developer profile) `server` (cluster and enterprise profiles)	Specifies whether the server instance is a regular instance (`server`), a domain administration server (`das`), or a combination (`das-and-server`). modifying this value is not recommended.
`system-jmx-connector-name`	none	Specifies the name of the internal "jmx-connector" on page 98.

alert-service

Configures the alert service, which allows you to register for and receive system status alerts.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `alert-service` element.

TABLE 1–7 `alert-service` Subelements

Element	Required	Description
"alert-subscription" on page 27	zero or more	Configures a subscription to system status alerts.
"property" on page 125	zero or more	Specifies a property or a variable.

alert-subscription

Configures a subscription to system status alerts.

Superelements

"alert-service" on page 26

Subelements

The following table describes subelements for the `alert-subscription` element.

TABLE 1–8 `alert-subscription` Subelements

Element	Required	Description
"listener-config" on page 104	only one	Configures the listener class that listens for alerts from notification emitters.
"filter-config" on page 70	zero or one	Configures the filter class that filters alerts from notification emitters.

Attributes

The following table describes attributes for the `alert-subscription` element.

TABLE 1–9 `alert-subscription` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of this alert subscription.

appclient-module

Specifies a deployed application client container (ACC) module.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `appclient-module` element.

TABLE 1–10 `appclient-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `appclient-module` element.

TABLE 1–11 `appclient-module` Attributes

Attribute	Default	Description
`name`	none	The name of the ACC module.
`location`	none	The location of the ACC module in the Communications Application Server file system.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.
`java-web-start-enabled`	`true`	(optional) Specifies whether Java Web Start access is permitted for this application client.

application-ref

References an application or module deployed to the server instance or cluster.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"cluster" on page 36, "server" on page 137

Subelements

none

Attributes

The following table describes attributes for the `application-ref` element.

TABLE 1–12 `application-ref` Attributes

Attribute	Default	Description
`enabled`	`true`	(optional) Determines whether the application or module is enabled.
`virtual-servers`	all virtual servers	(optional) In a comma-separated list, references `id` attributes of the "virtual-server" on page 162 elements to which the "web-module" on page 171 or the web modules within this "j2ee-application" on page 86 are deployed.
`lb-enabled`	`false`	(optional) If `true`, all load-balancers that reference this application consider it available to them.
`disable-timeout-in-minutes`	`30`	(optional) Specifies the time it takes this application to reach a quiescent state after having been disabled.
`ref`	none	References the `name` attribute of a "lifecycle-module" on page 103, "j2ee-application" on page 86, "ejb-module" on page 61, "web-module" on page 171, "connector-module" on page 45, "appclient-module" on page 27, or "extension-module" on page 67 element.

applications

Contains deployed Java EE applications, Java EE modules, and Lifecycle modules.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `applications` element.

TABLE 1–13 `applications` Subelements

Element	Required	Description
"lifecycle-module" on page 103	zero or more	Specifies a deployed lifecycle module.
"j2ee-application" on page 86	zero or more	Specifies a deployed Java EE application.
"ejb-module" on page 61	zero or more	Specifies a deployed EJB module.
"web-module" on page 171	zero or more	Specifies a deployed web module.
"connector-module" on page 45	zero or more	Specifies a deployed connector module.
"appclient-module" on page 27	zero or more	Specifies a deployed application client container (ACC) module.
"mbean" on page 113	zero or more	Specifies an MBean.
"extension-module" on page 67	zero or more	Specifies an extension module.

Note – Subelements of an `applications` element can occur in any order.

audit-module

Specifies an optional plug-in module that implements audit capabilities.

Superelements

"security-service" on page 136

Subelements

The following table describes subelements for the `audit-module` element.

TABLE 1–14 `audit-module` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `audit-module` element.

TABLE 1–15 `audit-module` Attributes

Attribute	Default	Description
`name`	`default`	Specifies the name of this audit module.
`classname`	`com.sun.enterprise.security.Audit`	Specifies the Java class that implements this audit module.

auth-realm

Defines a realm for authentication.

Authentication realms require provider-specific properties, which vary depending on what a particular implementation needs.

For more information about how to define realms, see the *Sun Java System Application Server 9.1 Administration Guide*.

Here is an example of the default `file` realm:

```
<auth-realm name="file"
     classname="com.sun.enterprise.security.auth.realm.file.FileRealm">
     <property name="file" value="domain-dir/config/keyfile"/>
     <property name="jaas-context" value="fileRealm"/>
 </auth-realm>
```

Which properties an `auth-realm` element uses depends on the value of the `auth-realm` element's `name` attribute. The `file` realm uses `file` and `jaas-context` properties. Other realms use different properties.

Superelements

"node-agent" on page 120, "security-service" on page 136

Subelements

The following table describes subelements for the `auth-realm` element.

TABLE 1–16 `auth-realm` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `auth-realm` element.

TABLE 1–17 `auth-realm` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of this realm.
`classname`	none	Specifies the Java class that implements this realm.

Properties

The standard realms provided with Communications Application Server have required and optional properties. A custom realm might have different properties.

The following table describes properties for the `auth-realm` element.

TABLE 1–18 `auth-realm` Properties

Property	Realms	Description
`jaas-context`	`file, ldap, jdbc, solaris`	Specifies the JAAS (Java Authentication and Authorization Service) context.
`file`	`file`	Specifies the file that stores user names, passwords, and group names. The default is *domain-dir*`/config/keyfile`.
`assign-groups`	`certificate, file, jdbc, ldap, solaris`	(optional) If this property is set, its value is taken to be a comma-separated list of group names. All clients who present valid certificates are assigned membership to these groups for the purposes of authorization decisions in the web and EJB containers.
`directory`	`ldap`	Specifies the LDAP URL to your server.
`base-dn`	`ldap`	Specifies the LDAP base DN for the location of user data. This base DN can be at any level above the user data, since a tree scope search is performed. The smaller the search tree, the better the performance.
`search-filter`	`ldap`	(optional) Specifies the search filter to use to find the user. The default is `uid=%s` (`%s` expands to the subject name).
`group-base-dn`	`ldap`	(optional) Specifies the base DN for the location of groups data. By default, it is same as the `base-dn`, but it can be tuned, if necessary.
`group-search-filter`	`ldap`	(optional) Specifies the search filter to find group memberships for the user. The default is `uniquemember=%d` (`%d` expands to the user element DN).
`group-target`	`ldap`	(optional) Specifies the LDAP attribute name that contains group name entries. The default is `CN`.
`search-bind-dn`	`ldap`	(optional) Specifies an optional DN used to authenticate to the directory for performing the `search-filter` lookup. Only required for directories that do not allow anonymous search.
`search-bind-password`	`ldap`	(optional) Specifies the LDAP password for the DN given in `search-bind-dn` .
`datasource-jndi`	`jdbc`	Specifies the `jndi-name` of the "jdbc-resource" on page 94 for the database.
`user-table`	`jdbc`	Specifies the name of the user table in the database.
`user-name-column`	`jdbc`	Specifies the name of the user name column in the database's user table.
`password-column`	`jdbc`	Specifies the name of the password column in the database's user table.
`group-table`	`jdbc`	Specifies the name of the group table in the database.
`group-name-column`	`jdbc`	Specifies the name of the group name column in the database's group table.
`db-user`	`jdbc`	(optional) Allows you to specify the database user name in the realm instead of the "jdbc-connection-pool" on page 90. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.

TABLE 1–18 `auth-realm` Properties *(Continued)*

Property	Realms	Description
`db-password`	`jdbc`	(optional) Allows you to specify the database password in the realm instead of the "jdbc-connection-pool" on page 90. This prevents other applications from looking up the database, getting a connection, and browsing the user table. By default, the `jdbc-connection-pool` configuration is used.
`digest-algorithm`	`jdbc`	(optional) Specifies the digest algorithm. The default is `MD5`. You can use any algorithm supported in the JDK, or `none`.
`encoding`	`jdbc`	(optional) Specifies the encoding. Allowed values are `Hex` and `Base64`. If `digest-algorithm` is specified, the default is `Hex`. If `digest-algorithm` is not specified, by default no encoding is specified.
`charset`	`jdbc`	(optional) Specifies the charset for the digest algorithm.

availability-service

Configures the availability service. Enables high-availability features, such as session state and stateful session bean state persistence.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Availability can be enabled or disabled at the following levels:

1. The server instance (attribute of `availability-service`). Default is `true` (enabled).
2. The EJB, SIP, or web container (attribute of "ejb-container-availability" on page 60, "sip-container-availability" on page 144, or "web-container-availability" on page 168). Default is `true` (enabled).
3. The application (attribute of "j2ee-application" on page 86). Default is `false` (disabled).
4. The stand-alone EJB, SIP, or web module (attribute of "ejb-module" on page 61, "extension-module" on page 67, or "web-module" on page 171). Default is `false` (disabled).
5. The stateful session bean. Default is `false` (disabled). See the *Sun Java System Application Server 9.1 Developer's Guide*.

For availability to be enabled at a given level, it must be enabled at all higher levels, as well. For example, to enable availability at the application level, you must also enable it at the server instance and container levels.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `availability-service` element.

TABLE 1–19 `availability-service` Subelements

Element	Required	Description
"web-container-availability" on page 168	zero or one	Enables availability in the web container.
"ejb-container-availability" on page 60	zero or one	Enables availability in the EJB container.
"sip-container-availability" on page 144	zero or one	Enables availability in the SIP container.
"converged-load-balancer" on page 51	zero or one	Defines and configures a converged load balancer.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `availability-service` element.

TABLE 1–20 `availability-service` Attributes

Attribute	Default	Description
`availability-enabled`	`true`	(optional) If set to `true`, high-availability features apply to all applications deployed to the server instance that do not have availability disabled. All instances in a cluster should have the same availability value to ensure consistent behavior.

Properties

The following table describes properties for the `availability-service` element. For more information about replicated session persistence, see "web-container-availability" on page 168 and "ejb-container-availability" on page 60.

TABLE 1–21 `availability-service` Properties

Attribute	Default	Description
`replication_measurement_enabled`	`false`	If `true`, logs measurements of replication times. One of these messages appears in the sending instance's log: `messageSendSucceeded: id =` *session-id* `fastAckTime = 8 to partner:` *instance-name* `messageSendFailed: id =` *session-id* `fastAckTime = 8 to partner:` *instance-name* This message appears in the receiving instance's log: `messageReceiptSucceeded: bulkId = 1 receiptTime = 12 from partner:` *instance-name*
`replication_measurement_interval`	`1`	Specifies the frequency of measurement of replication. It must be a positive integer: 1 means every replication, 2 means once every 2 replications, 3 means once every 3 replications, and so on. Applicable only if `replication_measurement_enabled` is set to `true`.

B

backend-principal

Specifies the user name and password required by the Enterprise Information System (EIS).

Superelements

"security-map" on page 135

Subelements

none

Attributes

The following table describes attributes for the `backend-principal` element.

TABLE 1–22 `backend-principal` Attributes

Attribute	Default	Description
`user-name`	none	Specifies the user name required by the EIS.
`password`	none	(optional) Specifies the password required by the EIS, if any.

C

cluster

Defines a cluster.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"clusters" on page 38

Subelements

The following table describes subelements for the `cluster` element.

TABLE 1–23 `cluster` Subelements

Element	Required	Description
"server-ref" on page 138	zero or more	References a server instance that belongs to the cluster.
"resource-ref" on page 132	zero or more	References a resource deployed to the cluster.
"application-ref" on page 28	zero or more	References an application or module deployed to the cluster.
"system-property" on page 154	zero or more	Specifies a system property.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `cluster` element.

TABLE 1–24 `cluster` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the cluster.

TABLE 1–24 `cluster` Attributes *(Continued)*

Attribute	Default	Description
`config-ref`	default "config" on page 38 element's `name` attribute value, `server-config`	References the configuration used by the cluster.
`heartbeat-port`	none; value automatically generated	Specifies the communication port the Group Management Service uses to listen for group events. Must be a valid port number.
`heartbeat-address`	none; value automatically generated	Specifies the address the Group Management Service uses to listen for group events. Must be a multicast address.
`heartbeat-enabled`	`false` (developer profile) `true` (cluster and enterprise profiles)	(optional) If `true`, the Group Management Service is started as a lifecycle module in each server instance in the cluster and in the Domain Administration Server. The Domain Administration Server participates in each cluster that has this attribute set to `true`.

cluster-ref

References a cluster.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"lb-config" on page 101

Subelements

The following table describes subelements for the `cluster-ref` element.

TABLE 1–25 `cluster-ref` Subelements

Element	Required	Description
"health-checker" on page 72	zero or one	Defines a health checker for the referenced cluster.

Attributes

The following table describes attributes for the `cluster-ref` element.

TABLE 1–26 `cluster-ref` Attributes

Attribute	Default	Description
`ref`	none	References the `name` attribute of a "cluster" on page 36 element.
`lb-policy`	`round-robin`	(optional) Specifies the load balancing policy. Allowed values are: ■ `round-robin` — The load balancer cycles through the cluster's server instances in a specified order. ■ `weighted-round-robin` — The load balancer cycles through the server instances in a specified order, but routes more requests to instances with greater processing capacity. Processing capacity is represented by values of the `lb-weight` attribute of each "server" on page 137 element referenced by the cluster. ■ `user-defined` — The load balancing policy is defined in a custom module.
`lb-policy-module`	none	(optional) Specifies the absolute path to the shared library that implements the user-defined policy. The shared library must exist and be readable on the machine where the load balancer is running. Required only if `lb-policy` is set to `user-defined`.

clusters

Contains clusters.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `clusters` element.

TABLE 1–27 `clusters` Subelements

Element	Required	Description
"cluster" on page 36	zero or more	Defines a cluster.

config

Defines a configuration, which is a collection of settings that controls how a server instance functions.

Superelements

"configs" on page 40

Subelements

The following table describes subelements for the `config` element.

TABLE 1–28 `config` Subelements

Element	Required	Description
"sip-service" on page 149	zero or one	Configures the SIP service.
"http-service" on page 79	only one	Configures the HTTP service.
"iiop-service" on page 84	only one	Configures the IIOP service.
"admin-service" on page 25	only one	Determines whether the server to which the configuration applies is an administration server.
"connector-service" on page 47	zero or one	Configures the connector service.
"web-container" on page 167	only one	Configures the web container.
"ejb-container" on page 58	only one	Configures the Enterprise JavaBeans™ (EJB™) container.
"mdb-container" on page 114	only one	Configures the message-driven bean (MDB) container.
"sip-container" on page 142	zero or one	Configures the SIP container.
"jms-service" on page 96	zero or one	Configures the Java Message Service (JMS) provider.
"log-service" on page 107	only one	Configures the system logging service.
"security-service" on page 136	only one	Configures the Java EE security service.
"transaction-service" on page 157	only one	Configures the transaction service.
"monitoring-service" on page 119	only one	Configures the monitoring service.
"diagnostic-service" on page 55	zero or one	Configures the diagnostic service.
"java-config" on page 88	only one	Configures the Java Virtual Machine (JVM™).
"availability-service" on page 33	zero or one	Configures the availability service.
"thread-pools" on page 157	only one	Configures thread pools.
"alert-service" on page 26	zero or one	Configures the alert service.
"group-management-service" on page 71	zero or one	Configures the group management service.
"management-rules" on page 111	zero or one	Configures self-management rules.
"system-property" on page 154	zero or more	Specifies a system property.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `config` element.

TABLE 1–29 `config` Attributes

Attribute	Default	Description
`name`	`server-config` (for default instance)	Specifies the name of the configuration.
`dynamic-reconfiguration-enabled`	`true`	(optional) If `true`, any changes to the system (for example, applications deployed, resources created) are automatically applied to the affected servers without a restart being required. If false, such changes are only picked up by the affected servers when each server restarts.

configs

Contains configurations.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `configs` element.

TABLE 1–30 `configs` Subelements

Element	Required	Description
"config" on page 38	only one (developer profile) one or more (cluster and enterprise profiles)	Defines a configuration.

connection-pool

Defines a pool of client HTTP or SIP connections used by the "http-listener" on page 74 or "sip-listener" on page 147 subelements of the parent "http-service" on page 79 or "sip-service" on page 149 element.

Superelements

"http-service" on page 79, "sip-service" on page 149

Subelements

none

Attributes

The following table describes attributes for the `connection-pool` element.

TABLE 1–31 `connection-pool` Attributes

Attribute	Default	Description
`queue-size-in-bytes`	`4096`	(optional) Specifies the size in bytes of the connection queue for "http-listener" on page 74 or "sip-listener" on page 147 elements.
`max-pending-count`	`4096`	(optional) Specifies the maximum number of pending connections on an "http-listener" on page 74 or "sip-listener" on page 147.
`receive-buffer-size-in-bytes`	`4096`	(optional) Specifies the size of the receive buffer for all "http-listener" on page 74 or "sip-listener" on page 147 elements.
`send-buffer-size-in-bytes`	`8092`	(optional) Specifies the size of the send buffer for all "http-listener" on page 74 or "sip-listener" on page 147 elements.

connector-connection-pool

Defines a connector connection pool.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `connector-connection-pool` element.

TABLE 1–32 `connector-connection-pool` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"security-map" on page 135	zero or more	Maps the principal received during servlet or EJB authentication to the credentials accepted by the EIS.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `connector-connection-pool` element. Changing the following attributes requires a server restart: `resource-adapter-name`, `connection-definition-name`, `transaction-support`, `associate-with-thread`, `lazy-connection-association`, and `lazy-connection-enlistment`.

TABLE 1–33 `connector-connection-pool` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connection pool. A "connector-resource" on page 46 element's `pool-name` attribute refers to this `name`.
`resource-adapter-name`	none	Specifies the `name` attribute of the deployed "connector-module" on page 45. If no name is specified during deployment, the name of the `.rar` file is used. If the resource adapter is embedded in an application, then it is *app_name#rar_name*.
`connection-definition-name`	none	Specifies a unique name, identifying a resource adapter's `connection-definition` element in the `ra.xml` file. This is usually the `connectionfactory-interface` of the `connection-definition` element.
`steady-pool-size`	`8`	(optional) Specifies the initial and minimum number of connections maintained in the pool.
`max-pool-size`	`32`	(optional) Specifies the maximum number of connections that can be created to satisfy client requests.
`max-wait-time-in-millis`	`60000`	(optional) Specifies the amount of time, in milliseconds, that the caller is willing to wait for a connection. If `0`, the caller is blocked indefinitely until a resource is available or an error occurs.
`pool-resize-quantity`	`2`	(optional) Specifies the number of idle connections to be destroyed if the existing number of connections is above the `steady-pool-size` (subject to the `max-pool-size` limit). This is enforced periodically at the `idle-timeout-in-seconds` interval. An idle connection is one that has not been used for a period of `idle-timeout-in-seconds`. When the pool size reaches `steady-pool-size`, connection removal stops.
`idle-timeout-in-seconds`	`300`	(optional) Specifies the maximum time that a connection can remain idle in the pool. After this amount of time, the pool can close this connection.
`fail-all-connections`	`false`	(optional) If `true`, closes all connections in the pool if a single validation check fails.
`transaction-support`	none	(optional) Specifies the transaction support for this connection pool. Overrides the transaction support defined in the resource adapter in a downward compatible way: supports a transaction level lower than or equal to the resource adapter's, but not higher. Allowed values in descending order are: ■ `XATransaction` - Supports distributed transactions. ■ `LocalTransaction` - Supports local transactions only. ■ `NoTransaction` - No transaction support.

TABLE 1–33 `connector-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`is-connection-validation-required`	`false`	(optional) Specifies whether connections have to be validated before being given to the application. If a resource's validation fails, it is destroyed, and a new resource is created and returned.
`connection-leak-timeout-in-seconds`	`0`	Detects potential connection leaks by the application. A connection that is not returned back to the pool by the application within the specified period is assumed to be potentially leaking, and a stack trace of the caller is logged. A zero value disables leak detection. A nonzero value enables leak tracing.
`connection-leak-reclaim`	`false`	If `true`, the pool will reclaim a connection after `connection-leak-timeout-in-seconds` occurs.
`connection-creation-retry-attempts`	`0`	Specifies the number of attempts to create a new connection.
`connection-creation-retry-interval-in-seconds`	`10`	Specifies the time interval between attempts to create a connection when `connection-creation-retry-attempts` is greater than `0`.
`validate-atmost-once-period-in-seconds`	`60`	Specifies the time interval within which a connection is validated at most once. Minimizes the number of validation calls.
`lazy-connection-enlistment`	`false`	If `true`, a connection is not enlisted in a transaction until it is used. If `false`, any connection object available to a transaction is enlisted in the transaction.
`lazy-connection-association`	`false`	If `true`, a physical connection is not associated with a logical connection until it is used. If `false`, a physical connection is associated with a logical connection even before it is used.
`associate-with-thread`	`false`	If `true`, allows a connection to be saved as a `ThreadLocal` in the calling thread. This connection gets reclaimed only when the calling thread dies or when the calling thread is not in use and the pool has run out of connections.
`match-connections`	`true`	If `true`, enables connection matching. You can set to `false` if connections are homogeneous.
`max-connection-usage-count`	`0`	Specifies the number of times a connections is reused by the pool, after which it is closed. A zero value disables this feature.

Properties

Most properties of the `connector-connection-pool` element are the names of setter methods of the `managedconnectionfactory-class` element in the `ra.xml` file. Properties of the `connector-connection-pool` element override the `ManagedConnectionFactory` JavaBean configuration settings.

All but the last four properties in the following table are `connector-connection-pool` properties of `jmsra`, the resource adapter used to communicate with the Sun Java System Message Queue software. For a complete list of the available properties (called *administered object attributes* in the Message Queue software), see the *Sun Java System Message Queue 3.7 UR1 Administration Guide*.

Changes to `connector-connection-pool` properties require a server restart.

TABLE 1–34 `connector-connection-pool` Properties

Property	Default	Description
`AddressList`	none	Specifies a list of host/port combinations of the Message Queue software. For JMS resources of the Type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`.
`ClientId`	none	Specifies the JMS Client Identifier to be associated with a `Connection` created using the `createTopicConnection` method of the `TopicConnectionFactory` class. For JMS resources of the Type `javax.jms.TopicConnectionFactory` . Durable subscription names are unique and only valid within the scope of a client identifier. To create or reactivate a durable subscriber, the connection must have a valid client identifier. The JMS specification ensures that client identifiers are unique and that a given client identifier is allowed to be used by only one active connection at a time.
`UserName`	`guest`	Specifies the user name for connecting to the Message Queue software. For JMS resources of the Type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`.
`Password`	`guest`	Specifies the password for connecting to the Message Queue software. For JMS resources of the Type `javax.jms.TopicConnectionFactory` or `javax.jms.QueueConnectionFactory`.
`ReconnectAttempts`	`6`	Specifies the number of attempts to connect (or reconnect) for each address in the `imqAddressList` before the client runtime moves on to try the next address in the list. A value of `-1` indicates that the number of reconnect attempts is unlimited (the client runtime attempts to connect to the first address until it succeeds).
`ReconnectInterval`	`30000`	Specifies the interval between reconnect attempts in milliseconds. This applies to attempts on each address in the `imqAddressList` and on successive addresses in the list. If too short, this time interval does not give a broker time to recover. If too long, the reconnect might represent an unacceptable delay.
`ReconnectEnabled`	`false`	If `true`, specifies that the client runtime attempts to reconnect to a message server (or the list of addresses in `imqAddressList`) when a connection is lost.
`AddressListBehavior`	`priority`	Specifies whether connection attempts are in the order of addresses in the `imqAddressList` attribute (`priority`) or in a random order (`random`). If many clients are attempting a connection using the same connection factory, use a random order to prevent them from all being connected to the same address.
`AddressListIterations`	`-1`	Specifies the number of times the client runtime iterates through the `imqAddressList` in an effort to establish (or reestablish) a connection. A value of `-1` indicates that the number of attempts is unlimited.
`LazyConnection Enlistment`	`false`	Deprecated. Use the equivalent attribute.
`LazyConnection Association`	`false`	Deprecated. Use the equivalent attribute.
`AssociateWithThread`	`false`	Deprecated. Use the equivalent attribute.

TABLE 1–34 `connector-connection-pool` Properties *(Continued)*

Property	Default	Description
`MatchConnections`	`true`	Deprecated. Use the equivalent attribute.

Note – All JMS administered object resource properties that worked with version 7 of the Communications Application Server are supported for backward compatibility.

connector-module

Specifies a deployed connector module.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `connector-module` element.

TABLE 1–35 `connector-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `connector-module` element.

TABLE 1–36 `connector-module` Attributes

Attribute	Default	Description
`name`	none	The name of the connector module.
`location`	none	The location of the connector module in the Communications Application Server file system.

TABLE 1–36 `connector-module` Attributes *(Continued)*

Attribute	Default	Description
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the connector module is enabled.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.

connector-resource

Defines the connection factory object of a specific connection definition in a connector (resource adapter).

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `connector-resource` element.

TABLE 1–37 `connector-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `connector-resource` element.

TABLE 1–38 `connector-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.

TABLE 1–38 `connector-resource` Attributes *(Continued)*

Attribute	Default	Description
`pool-name`	none	Specifies the `name` of the associated connector connection pool, defined in a "connector-connection-pool" on page 41 element.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

connector-service

Configures the connector service.

Superelements

"config" on page 38

Subelements

none

Attributes

The following table describes attributes for the `connector-service` element.

TABLE 1–39 `connector-service` Attributes

Attribute	Default	Description
`shutdown-timeout-in-seconds`	30	(optional) Specifies the maximum time allowed during application server shutdown for the `ResourceAdapter.stop()` method of a connector module's instance to complete. Resource adapters that take longer to shut down are ignored, and Communications Application Server shutdown continues.

converged-lb-cluster-ref

References a target cluster to which requests are forwarded by a converged load balancer.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"converged-lb-config" on page 48

Subelements

none

Attributes

The following table describes attributes for the `converged-lb-cluster-ref` element.

TABLE 1–40 `converged-lb-cluster-ref` Attributes

Attribute	Default	Description
`ref`	none	References the `name` attribute of a "cluster" on page 36 element.
`self-loadbalance`	`true`	(optional) If `true`, specifies that a configured cluster load balances incoming requests to itself. If `true`, the parent "converged-lb-config" on page 48 element must have exactly one `converged-lb-cluster-ref` subelement. In this case, the load balancer is an intrinsic component of the participating server instances in the cluster. For production environments, only the `true` setting is supported.

converged-lb-config

Defines a converged load balancer configuration, which can be referenced by a "converged-load-balancer" on page 51.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"converged-lb-configs" on page 49

Subelements

The following table describes subelements for the `converged-lb-config` element.

TABLE 1–41 `converged-lb-config` Subelements

Element	Required	Description
"converged-lb-policy" on page 50	only one	Specifies the load balancing policy used by the converged load balancer.
"converged-lb-cluster-ref" on page 47	one or more; only one if this subelement's `selfloadbalance` attribute is `true`; zero if a `server-ref` is defined	References a cluster. This element contains some attributes related to load balancing.
"server-ref" on page 138	one or more; zero if a `converged-lb-cluster-ref` is defined	References a stand-alone server instance, which does not belong to a cluster. The referenced "server" on page 137 element contains some attributes related to load balancing.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `converged-lb-config` element.

TABLE 1–42 `converged-lb-config` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the converged load balancer configuration.

converged-lb-configs

Contains converged load balancer configurations.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `converged-lb-configs` element.

TABLE 1–43 `converged-lb-configs` Subelements

Element	Required	Description
"converged-lb-config" on page 48	zero or more	Defines a converged load balancer configuration.

converged-lb-policy

Specifies the load balancing policy used by the converged load balancer.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"converged-lb-config" on page 48

Subelements

The following table describes subelements for the `converged-lb-policy` element.

TABLE 1–44 `converged-lb-policy` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `converged-lb-policy` element.

TABLE 1–45 `converged-lb-policy` Attributes

Attribute	Default	Description
`http`	`round-robin`	(optional) Specifies the load balancing policy used for HTTP requests. The only allowed value is `round-robin`, which means that the load balancer cycles through the cluster's server instances in a specified order.
`sip`	`from-tag,to-tag, call-id`	(optional) Specifies the parameters on which a consistent hashing policy is applied to obtain the hash key. This can be a single value or comma-separated values of parameter names to hash on. If more than one parameter is specified, the concatenated values of the parameters are used for applying the consistent hashing.

TABLE 1–45 `converged-lb-policy` Attributes *(Continued)*

Attribute	Default	Description
`dcr-file`	none	(optional) Specifies the `data-centric-rules.xml` file path name, which provides the rules for applying consistent hashing on both HTTP and SIP requests. By default this file is not specified. If specified, this file takes precedence over the `http` and `sip` attributes. This can be an absolute or relative path. A relative path is relative to *domain-dir*`/`*cluster*`/config`. If an HTTP request doesn't match any DCR file rules, a hash key is generated using the remote host and port. If a SIP request doesn't match any DCR file rules, a hash key is generated using `from-tag,to-tag,call-id`.

converged-load-balancer

Defines and configures a converged load balancer. For more information about load balancing in the Communications Application Server, see the *Sun Java System Application Server 9.1 High Availability Administration Guide*.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"availability-service" on page 33

Subelements

The following table describes subelements for the `converged-load-balancer` element.

TABLE 1–46 `converged-load-balancer` Subelements

Element	Required	Description
"proxy" on page 128	only one	Specifies the proxy configuration for the converged load balancer.

Attributes

The following table describes attributes for the `converged-load-balancer` element.

TABLE 1–47 `converged-load-balancer` Attributes

Attribute	Default	Description
`name`	none	Specifies a unique name for the load balancer.
`config-file`	*domain-dir*`/`*cluster*`/config/converged-loadbalancer.xml`	Specifies the load balancer's configuration file, `converged-loadbalancer.xml`. This can be an absolute or relative path. A relative path is relative to *domain-dir*`/`*cluster*`/config`.
`auto-commit`	`false`	(optional) If `true`, configuration file changes are propagated to load balancer instances immediately.
`converged-lb-config-name`	none	Specifies the `name` of the "converged-lb-config" on page 48 used by the load balancer.

custom-resource

Defines a custom resource, which specifies a custom server-wide resource object factory. Such object factories implement the `javax.naming.spi.ObjectFactory` interface.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `custom-resource` element.

TABLE 1–48 `custom-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `custom-resource` element.

TABLE 1–49 `custom-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`res-type`	none	Specifies the fully qualified type of the resource.

TABLE 1–49 `custom-resource` Attributes *(Continued)*

Attribute	Default	Description
`factory-class`	none	Specifies the fully qualified name of the user-written factory class, which implements `javax.naming.spi.ObjectFactory`.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

D

das-config

Defines a domain administration server configuration. The domain administration server runs the Administration Console.

Superelements

"admin-service" on page 25

Subelements

The following table describes subelements for the `das-config` element.

TABLE 1–50 `das-config` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `das-config` element. For more information about deployment topics such as dynamic reloading and autodeployment, see the *Sun Java System Application Server 9.1 Developer's Guide*.

TABLE 1–51 `das-config` Attributes

Attribute	Default	Description
`dynamic-reload-enabled`	`false`	(optional) If `true`, checks the timestamp on a `.reload` file at every module and application directory level, to trigger dynamic reloading.
`dynamic-reload-poll-interval-in-seconds`	`2`	(optional) Controls the polling frequency of dynamic reloading.
`autodeploy-enabled`	`false`	(optional) If `true`, enables autodeployment, which lets you quickly deploy applications and modules to a running Communications Application Server without performing an explicit server instance restart or a separate deployment operation.
`autodeploy-polling-interval-in-seconds`	`2`	(optional) Controls the polling frequency of autodeployment.
`autodeploy-dir`	`autodeploy`	(optional) Specifies the source directory (absolute or relative to *domain-dir*) in which autodeployment looks for deployable components.
`autodeploy-verifier-enabled`	`false`	(optional) If `true`, the verifier is run before autodeployment. If verification fails, deployment is not performed.
`autodeploy-jsp-precompilation-enabled`	`false`	(optional) If `true`, JSP pages are precompiled during autodeployment.
`deploy-xml-validation`	`full`	(optional) Specifies the type of XML validation performed on standard and Communications Application Server deployment descriptors: ■ `full` - If XML validation fails, deployment fails. ■ `parsing` - XML validation errors are reported but deployment occurs. ■ `none` - No XML validation is performed.
`admin-session-timeout-in-minutes`	`sun-web.xml` `timeoutSeconds` property value or `web.xml` `session-timeout` attribute value	(optional) Specifies the Administration Console timeout.

description

Contains a text description of the parent element.

Superelements

"admin-object-resource" on page 24, "appclient-module" on page 27, "connector-connection-pool" on page 41, "connector-module" on page 45, "connector-resource" on page 46, "custom-resource" on page 52, "ejb-module" on page 61, "event" on page 63, "extension-module" on page 67, "external-jndi-resource" on page 69, "j2ee-application" on page 86, "jdbc-connection-pool" on page 90, "jdbc-resource" on page 94, "lifecycle-module" on page 103, "mail-resource" on page 108, "management-rule" on page 110, "mbean" on page 113, "persistence-manager-factory-resource" on page 123, "property" on page 125, "system-property" on page 154, "transformation-rule" on page 159, "web-module" on page 171

Subelements

none - contains data

diagnostic-service

Configures the Diagnostic Service, which lets you generate a diagnostic report for troubleshooting in case of Communications Application Server malfunctioning such as exceptions, performance bottlenecks, or unexpected results.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `diagnostic-service` element.

TABLE 1–52 `diagnostic-service` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `diagnostic-service` element.

TABLE 1–53 `diagnostic-service` Attributes

Attribute	Default	Description
`compute-checksum`	`true`	(optional) If `true`, computes a checksum of binaries.

TABLE 1–53 `diagnostic-service` Attributes *(Continued)*

Attribute	Default	Description
`verify-config`	`true`	(optional) If `true`, captures the output of the `asadmin verify-domain-xml` command.
`capture-install-log`	`true`	(optional) If `true`, captures the log generated during Communications Application Server installation.
`capture-system-info`	`true`	(optional) If `true`, collects operating system level information.
`capture-app-dd`	`true`	(optional) If `true`, captures application deployment descriptors in plain text. If any deployment descriptors contain confidential information, you should set it to `false`.
`min-log-level`	`INFO`	(optional) Specifies the log level for the diagnostic report. See "module-log-levels" on page 116 for a description of log levels. If set to `OFF`, log contents are not captured.
`max-log-entries`	`500`	(optional) Specifies the maximum number of log entries captured.

domain

Defines a domain. This is the root element; there can only be one `domain` element in a `domain.xml` file.

Superelements

none

Subelements

The following table describes subelements for the `domain` element.

TABLE 1–54 `domain` Subelements

Element	Required	Description
"applications" on page 29	zero or one	Contains deployed Java EE applications, Java EE modules, and lifecycle modules.
"resources" on page 133	zero or one	Contains configured resources.
"configs" on page 40	only one	Contains configurations.
"servers" on page 139	only one	Contains server instances.
"clusters" on page 38	zero or one	Contains clusters.
"node-agents" on page 121	zero or one	Contains node agents.
"lb-configs" on page 102	zero or one	Contains load balancing configurations.

TABLE 1–54 `domain` Subelements *(Continued)*

Element	Required	Description
"converged-lb-configs" on page 49	zero or one	Contains converged load balancing configurations.
"load-balancers" on page 106	zero or one	Contains load balancers.
"system-property" on page 154	zero or more	Specifies a system property.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `domain` element.

TABLE 1–55 `domain` Attributes

Attribute	Default	Description
`application-root`	*domain-dir*`/applications`	(optional) Specifies the absolute path where deployed applications reside for this domain.
`log-root`	*domain-dir*`/logs`	(optional) Specifies where the domain's log files are kept. The directory in which the log is kept must be writable by whatever user account the server runs as. See the "log-service" on page 107 description for details about logs.
`locale`	operating system default	(optional) Specifies the domain's language.

host-name

Identifies the trusted host on the network using a domain name, for example, `sun.com` or `company22.com`.

Superelements

"trusted-entity" on page 160

Subelements

none - contains data

E

ejb-container

Configures the EJB container. Stateless session beans are maintained in pools. Stateful session beans have session affinity and are cached. Entity beans associated with a database primary key are also cached. Entity beans not yet associated with a primary key are maintained in pools. Pooled entity beans are used to run `ejbCreate()` and finder methods.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `ejb-container` element.

TABLE 1–56 `ejb-container` Subelements

Element	Required	Description
"ejb-timer-service" on page 62	zero or one	Configures the EJB timer service.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `ejb-container` element.

TABLE 1–57 `ejb-container` Attributes

Attribute	Default	Description
`steady-pool-size`	`32`	(optional) Specifies the initial and minimum number of beans maintained in the pool. Must be `0` or greater and less than `max-pool-size` . Bean instances are removed from the pool and returned after use. The pool is replenished or cleaned up periodically to maintain this size. Applies to stateless session beans and entity beans.

TABLE 1–57 `ejb-container` Attributes *(Continued)*

Attribute	Default	Description
`pool-resize-quantity`	`16`	(optional) Specifies the number of beans to be removed when the `pool-idle-timeout-in-seconds` timer expires. A cleaner thread removes any unused instances. Must be `0` or greater and less than `max-pool-size` . The pool is not resized below the `steady-pool-size`. Applies to stateless session beans and entity beans.
`max-pool-size`	`64`	(optional) Specifies the maximum number of beans that can be created to satisfy client requests. A value of `0` indicates an unbounded pool. Applies to stateless session beans and entity beans.
`cache-resize-quantity`	`32`	(optional) Specifies the number of beans to be: ■ created if a request arrives when the pool has no available beans (subject to the `max-cache-size` limit) ■ passivated when the `cache-idle-timeout-in-seconds` timer expires and a cleaner thread removes any unused instances, or when the cache size exceeds `max-cache-size`. Must be greater than `1` and less than `max-cache-size`. Applies to stateful session beans and entity beans.
`max-cache-size`	`512`	(optional) Specifies the maximum number of beans in the cache. A value of `0` indicates an unbounded cache. Applies to stateful session beans and entity beans.
`pool-idle-timeout-in-seconds`	`600`	(optional) Specifies the maximum time that a bean can remain idle in the pool. After this amount of time, the pool can remove this bean. A value of `0` specifies that idle beans can remain in the pool indefinitely. Applies to stateless session beans and entity beans.
`cache-idle-timeout-in-seconds`	`600`	(optional) Specifies the maximum time that a bean can remain idle in the cache. After this amount of time, the container can passivate this bean. A value of `0` specifies that beans never become candidates for passivation. Applies to stateful session beans and entity beans.
`removal-timeout-in-seconds`	`5400`	(optional) Specifies the amount of time that a bean can remain passivated before it is removed from the session store. A value of `0` specifies that the container does not remove inactive beans automatically. If `removal-timeout-in-seconds` is less than or equal to `cache-idle-timeout-in-seconds`, beans are removed immediately without being passivated. The `session-store` attribute of the "server" on page 137 element determines the location of the session store. Applies to stateful session beans.

TABLE 1–57 `ejb-container` Attributes *(Continued)*

Attribute	Default	Description
`victim-selection-policy`	`nru`	(optional) Specifies how stateful session beans are selected for passivation. Allowed values are `fifo`, `lru`, and `nru`: ■ `fifo` - Selects the oldest instance. ■ `lru` - Selects the least recently accessed instance. ■ `nru` - Selects a not recently used instance.
`commit-option`	`B`	(optional) Determines which commit option is used for entity beans. Legal values are `B` or `C`.
`session-store`	*domain-dir*`/session-store`	(optional) Specifies the directory where passivated stateful session beans and persisted HTTP sessions are stored in the file system.

ejb-container-availability

Enables availability in the EJB container, including stateful session bean (SFSB) state persistence. For additional replicated session persistence properties you can set, see "availability-service" on page 33.

Superelements

"availability-service" on page 33

Subelements

The following table describes subelements for the `ejb-container-availability` element.

TABLE 1–58 `ejb-container-availability` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `ejb-container-availability` element.

TABLE 1–59 `ejb-container-availability` Attributes

Attribute	Default	Description
`availability-enabled`	`true`	(optional) If set to `true`, and if availability is enabled for the server instance (see "availability-service" on page 33), high-availability features apply to all SFSBs deployed to the server instance that do not have availability disabled. All instances in a cluster should have the same availability value to ensure consistent behavior.
`sfsb-ha-persistence-type`	`ha`	(optional) Specifies the session persistence and passivation mechanism for SFSBs that have availability enabled. Allowed values are `file` (the file system) and `replicated` (other servers). If set to `file`, the "ejb-container" on page 58 element's `session-store` attribute specifies the file system location where the passivated session bean state is stored. Checkpointing to the file system is useful for internal testing but is not supported for production environments.
`sfsb-persistence-type`	`file`	(optional) Specifies the passivation mechanism for SFSBs that *do not* have availability enabled. Allowed values are `file` and `replicated`.

ejb-module

Specifies a deployed EJB module.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `ejb-module` element.

TABLE 1–60 `ejb-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 172	zero or more	Configures a web service endpoint.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `ejb-module` element.

TABLE 1–61 `ejb-module` Attributes

Attribute	Default	Description
`name`	none	The name of the EJB module.
`location`	none	The location of the EJB module in the Communications Application Server file system.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the EJB module is enabled.
`libraries`	none	(optional) Specifies an absolute or relative path to libraries specific to this module or application. A relative path is relative to *domain-dir*`/lib/applibs`. If the path is absolute, the path must be accessible to the domain administration server (DAS), which means it must be under *domain-dir*. To include more than one path, use a system-specific separator, such as a colon for Solaris or a semicolon for Windows. The libraries are made available to the application in the order in which they are specified.
`availability-enabled`	`false`	(optional) Specifies whether availability is enabled in this EJB module for SFSB checkpointing (and potentially passivation). Availability must also be enabled for the application or stand-alone EJB module during deployment. For more information about availability, see "availability-service" on page 33.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.

ejb-timer-service

Configures the EJB timer service.

Superelements

"ejb-container" on page 58

Subelements

The following table describes subelements for the `ejb-timer-service` element.

TABLE 1–62 `ejb-timer-service` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `ejb-timer-service` element.

TABLE 1–63 `ejb-timer-service` Attributes

Attribute	Default	Description
`minimum-delivery-interval-in-millis`	`7000`	(optional) Specifies the minimum time before an expiration for a particular timer can occur. This guards against extremely small timer increments that can overload the server.
`max-redeliveries`	`1`	(optional) Specifies the maximum number of times the EJB timer service attempts to redeliver a timer expiration due for exception or rollback.
`timer-datasource`	`jdbc/__TimerPool`	(optional) Overrides, for the cluster or server instance, the `cmp-resource` value specified in `sun-ejb-jar.xml` for the timer service system application (`__ejb_container_timer_app`).
`redelivery-interval-internal-in-millis`	`5000`	(optional) Specifies how long the EJB timer service waits after a failed `ejbTimeout` delivery before attempting a redelivery.

event

Defines the event that triggers the action associated with a management rule.

Predefined events are provided with the Communications Application Server. You can configure these events by changing `event` element attributes and properties.

You can create custom events by creating custom MBeans that implement the JMX `NotificationEmitter` interface. For more information about MBeans, see the *Sun Java System Application Server 9.1 Developer's Guide* and `http://java.sun.com/j2se/1.5.0/docs/api/javax/management/package-summary.html`. For information about monitor MBeans, see `http://java.sun.com/j2se/1.5.0/docs/api/javax/management/monitor/package-summary.html`.

Note – If multiple rules are associated with the same event, ordering of action execution is not guaranteed.

Superelements

"management-rule" on page 110

Subelements

The following table describes subelements for the `event` element.

TABLE 1–64 `event` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `event` element.

TABLE 1–65 `event` Attributes

Attribute	Default	Description
`type`	none	Specifies the type of event that triggers the management rule's action. Allowed values are as follows. The `cluster`, `lifecycle`, `log`, `monitor`, `timer`, and `trace` types are predefined events provided by the Communications Application Server. **Note –** Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*. ■ `cluster` — A Group Management Service (GMS) event. For more information, see "group-management-service" on page 71. ■ `lifecycle` — A lifecycle event. For more information about the server life cycle, see the *Sun Java System Application Server 9.1 Developer's Guide*. ■ `log` — An event in the server log. For more information about the server log, see "log-service" on page 107. ■ `monitor` — A monitoring event, which is a change in the attribute of a monitored "mbean" on page 113. ■ `notification` — A JMX notification event. Any custom "mbean" on page 113 that implements the JMX `NotificationEmitter` interface can be a notification event. ■ `timer` — An event that occurs at a specified time. ■ `trace` — A trace event. For descriptions of required and optional properties corresponding to each of these types, see the following table.
`level`	`INFO`	(optional) Specifies the level at which to log the event occurrence. For information about log levels, see "module-log-levels" on page 116. Applicable only if `record-event` is set to `true`.
`record-event`	`true`	(optional) Specifies whether the occurrence of the event is logged. If no "action" on page 23 is specified for the parent "management-rule" on page 110, the event is logged regardless of this setting. **Note –** Setting the `type` to `log` is different from setting `record-event` to `true`. The former specifies what the event is. The latter specifies what happens after the event occurs.

Properties

The following table describes properties for the `event` element. Property names are case insensitive.

TABLE 1–66 `event` Properties

Event Type	Property	Values	Description
`cluster`	`name`	`start`, `stop`, `fail`	Notifies when the GMS starts, stops, or reports failure of a server instance.
`cluster`	`serverName`	A comma-separated list of server instance names, or * for all server instances	Specifies the server instances about which GMS notifications are given.
`lifecycle`	`name`	`ready`, `shutdown`, `termination`	Specifies a server life cycle event. Values correspond to events defined in the `com.sun.appserv.server.LifecycleEvent` interface.
`log`	`loggerNames`	A comma-separated list of logger names, or * for all loggers, which is the default	(optional) Notifies when the specified loggers write messages to the server log. For a list of logger names, see "module-log-levels" on page 116.
`log`	`level`	A comma-separated list of log levels	(optional) Notifies when messages of the specified level are written to the server log. For information about log levels, see "module-log-levels" on page 116.
`monitor`	`observedMbean`	A `name` attribute of a user-defined "mbean" on page 113, or a JMX `ObjectName` for a system `mbean`	Specifies the name of the monitored MBean. Either this property or `observedObject` must be specified.
`monitor`	`observedObject`	An `object-name` attribute of a user-defined "mbean" on page 113, or a JMX `ObjectName` for a system `mbean`	Specifies the name of the monitored MBean. Either this property or `observedMbean` must be specified.
`monitor`	`observedAttribute`	An "mbean" on page 113 `Attribute` name	Specifies the monitored attribute of the monitored MBean.
`monitor`	`monitorType`	`CounterMonitor`, `GaugeMonitor`, `StringMonitor`	The type of monitoring of the attribute.
`monitor`	`granularityPeriod`	Time interval in seconds (`long int`)	(optional) Specifies the granularity at which the monitoring data should be collected.
`monitor`	`notifyMatch`	`true` or `false`	Specifies that the attribute value must match the `stringToCompare` value. Either this property or `notifyDiffer` is required if the monitor type is `StringMonitor`.

TABLE 1–66 event Properties *(Continued)*

Event Type	Property	Values	Description
monitor	notifyDiffer	true or false	Specifies that the attribute value must not match the stringToCompare value. Either this property or notifyMatch is required if the monitor type is StringMonitor.
monitor	stringToCompare	A String	Specifies the value to which the attribute value is compared. Required if the monitor type is StringMonitor.
monitor	numberType	byte, double, float, int, long, short	Specifies the type of the numeric value being monitored. Required if the monitor is of type CounterMonitor or GaugeMonitor.
monitor	differenceMode	true or false	Specifies the difference mode flag value common to all observed MBeans. Required if the monitor is of type CounterMonitor or GaugeMonitor.
monitor	initThreshold	A positive number of the type specified by numberType	Specifies a value above which notification occurs. Required if the monitor is of type CounterMonitor.
monitor	offset	A positive number of the type specified by numberType	(optional) Specifies that the event should be re-triggered when the initThreshold value plus this offset value is reached. Applicable if the monitor is of type CounterMonitor.
monitor	modulus	A positive number of the type specified by numberType	(optional) Specifies the modulus value common to all observed MBeans. Applicable if the monitor is of type CounterMonitor.
monitor	highThreshold	A positive number of the type specified by numberType	Specifies the upper limit of the range within which notification occurs. Required if the monitor is of type GaugeMonitor.
monitor	lowThreshold	A positive number of the type specified by numberType	Specifies the lower limit of the range within which notification occurs. Required if the monitor is of type GaugeMonitor.
notification	sourceMBean	name of "mbean" on page 113	Specifies a custom MBean that implements the JMX NotificationEmitter interface. Either this property or sourceObjectName must be specified.
notification	sourceObjectName	object-name of "mbean" on page 113	Specifies a custom MBean that implements the JMX NotificationEmitter interface. Either this property or sourceMBean must be specified.
notification	type	The notification type	(optional) Specifies the notification type. If this property is specified, the action of the parent "management-rule" on page 110 is performed only if the notification type emitted is same as this property's value.

TABLE 1–66 event Properties *(Continued)*

Event Type	Property	Values	Description
timer	dateString	Input format determined by pattern property	Begins notification at the specified date and time.
timer	pattern	SimpleDateFormat pattern	(optional) Specifies the date and time input format. The default is mm/dd/yyyy hh:mm:ss.
timer	period	Time interval in milliseconds (long int)	(optional) Notification repeats at the specified time interval.
timer	numberOfOccurrences	A positive number (long int)	(optional) Specifies the number of times notification occurs.
timer	message	A String	(optional) Specifies a message that is delivered as part of timer notification.
trace	name	web_component_method_entry, web_component_method_exit, ejb_component_method_entry, ejb_component_method_exit, request_start, request_end	Notifies at the specified trace point.
trace	ipAddress	An IP address	Specifies the IP address for which trace notifications are sent.
trace	callerPrincipal	A String	Specifies the caller principal for which trace notifications are sent.
trace	componentName	A String	Specifies the component name for which trace notifications are sent.

extension-module

Specifies a deployed extension module.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the extension-module element.

TABLE 1–67 `extension-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `extension-module` element.

TABLE 1–68 `extension-module` Attributes

Attribute	Default	Description
`name`	none	The name of the extension module.
`location`	none	The location of the extension module in the Communications Application Server file system.
`module-type`	none	Specifies a `String` that identifies the extension module type, which the runtime uses to find the appropriate add-on container. When an extension module is registered with the Communications Application Server, the Communications Application Server specifies the module type automatically. For example, the module type for a Session Initiation Protocol (SIP) servlet is `com.sun.enterprise.deployment.backend.extensions.sip.SipArchiveDeployer`.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the extension module is enabled.
`libraries`	none	(optional) Specifies an absolute or relative path to libraries specific to this module or application. A relative path is relative to *domain-dir*`/lib/applibs`. If the path is absolute, the path must be accessible to the domain administration server (DAS), which means it must be under *domain-dir*. To include more than one path, use a system-specific separator, such as a colon for Solaris or a semicolon for Windows. The libraries are made available to the application in the order in which they are specified.
`availability-enabled`	`false`	(optional) Specifies whether availability is enabled in this extension module. Availability must also be enabled for the application or stand-alone extension module during deployment. For more information about availability, see "availability-service" on page 33.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.

Properties

The following table describes properties for the `extension-module` element.

TABLE 1–69 `extension-module` Properties

Property	Default	Description
`context-root`	none	The context root at which the SIP or converged web/SIP extension module is deployed. The context root can be the empty string or just `/`. The context root can start with the `/` character, but doesn't have to. For load balancing to work, SIP or converged web/SIP extension module context roots must be unique within a cluster. See the *Sun Java System Application Server 9.1 High Availability Administration Guide* for more information about load balancing. **Note –** Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.
`isConverged`	`false`	If `true`, specifies that the SIP extension module contains both SIP and HTTP servlets and is converged. If `false`, specifies that the SIP extension module contains only SIP servlets.

external-jndi-resource

Defines a resource that resides in an external JNDI repository. For example, a generic Java object could be stored in an LDAP server. An external JNDI factory must implement the `javax.naming.spi.InitialContextFactory` interface.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `external-jndi-resource` element.

TABLE 1–70 `external-jndi-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `external-jndi-resource` element.

TABLE 1–71 `external-jndi-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`jndi-lookup-name`	none	Specifies the JNDI lookup name for the resource.
`res-type`	none	Specifies the fully qualified type of the resource.
`factory-class`	none	Specifies the fully qualified name of the factory class, which implements `javax.naming.spi.InitialContextFactory`. For more information about JNDI, see the *Sun Java System Application Server 9.1 Developer's Guide*.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

F

filter-config

Configures the filter class that filters alerts from notification emitters. See also "listener-config" on page 104.

Superelements

"alert-subscription" on page 27

Subelements

The following table describes subelements for the `filter-config` element.

TABLE 1–72 `filter-config` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `filter-config` element.

TABLE 1–73 `filter-config` Attributes

Attribute	Default	Description
`filter-class-name`	none	Specifies the class name of the filter.

G

group-management-service

Configures the Group Management Service (GMS), an in-process service that provides cluster monitoring, cluster membership, and group communication services. The GMS performs the following functions:

- Notifies registered modules in an Communications Application Server instance when one or more member instances are added to or removed from a cluster, or are suspected or confirmed to have failed.
- Provides the ability to send and receive messages between a group of processes.

The GMS is built atop a configurable stack of group membership discovery and health monitoring protocols. These protocols have properties that can be changed for a given network and deployment topology. Protocols in the GMS are as follows:

- Failure detection protocol - enables its members to periodically monitor other group members to determine their availability in the group.
- Discovery protocol - is used for discovery of the group and its members.
- Verify failure protocol - verifies suspect instances by adding a verification layer to mark a failure suspicion as a confirmed failure.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `group-management-service` element.

TABLE 1–74 `group-management-service` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `group-management-service` element.

TABLE 1–75 `group-management-service` Attributes

Attribute	Default	Description
`failure-detection-max-tries`	`3`	(optional) Specifies the maximum number of monitoring attempts before the GMS confirms that a failure is suspected in the group.
`failure-detection-timeout-in-millis`	`2000`	(optional) Specifies the time between monitoring attempts.
`discovery-timeout-in-millis`	`2000`	(optional) Specifies the time that the GMS waits for discovery of other members in the group.
`verify-failure-timeout-in-millis`	`1500`	(optional) Specifies the timeout after which a suspected failure is marked as verified.

H

health-checker

Defines a health checker for the parent "server-ref" on page 138 or "cluster-ref" on page 37 element.

Superelements

"cluster-ref" on page 37, "server-ref" on page 138

Subelements

none

Attributes

The following table describes attributes for the `health-checker` element.

TABLE 1–76 `health-checker` Attributes

Attribute	Default	Description
`url`	`/`	Specifies the URL to ping to determine the health state of a listener. This must be a relative URL.
`interval-in-seconds`	`30`	Specifies the interval between health checks. A value of zero means that health checking is disabled.
`timeout-in-seconds`	`10`	Specifies the maximum time in which a server must respond to a health check request to be considered healthy. If `interval-in-seconds` is greater than zero, `timeout-in-seconds` must be less than or equal to `interval-in-seconds`.

http-access-log

Defines an access log file for a "virtual-server" on page 162. The "access-log" on page 22 subelement of the virtual server's parent "http-service" on page 79 element determines the access log file's format and rotation settings.

Superelements

"virtual-server" on page 162

Subelements

none

Attributes

The following table describes attributes for the `http-access-log` element.

TABLE 1–77 `http-access-log` Attributes

Attribute	Default	Description
`log-directory`	`${com.sun.aas.instanceRoot}/logs/access`	(optional) Specifies the location of the access log file.
`iponly`	`true`	(optional) If `true`, specifies that only the IP address of the user agent is listed. If `false`, performs a DNS lookup.

http-file-cache

Configures the HTTP file cache.

Superelements

"http-service" on page 79

Subelements

none

Attributes

The following table describes attributes for the `http-file-cache` element.

TABLE 1–78 `http-file-cache` Attributes

Attribute	Default	Description
`globally-enabled`	`false` (developer profile) `true` (cluster and enterprise profiles)	(optional) If `true`, enables the file cache.
`file-caching-enabled`	`false` (developer profile) `true` (cluster and enterprise profiles)	(optional) If `true`, enables caching of the file content if the file size exceeds the `small-file-size-limit-in-bytes`.
`max-age-in-seconds`	`30`	(optional) Specifies the maximum age of a file cache entry.
`medium-file-size-limit-in-bytes`	`537600`	(optional) Specifies the maximum size of a file that can be cached as a memory mapped file.
`medium-file-space-in-bytes`	`10485760`	(optional) Specifies the total size of all files that are cached as memory mapped files.
`small-file-size-limit-in-bytes`	`2048`	(optional) Specifies the maximum size of a file that can be read into memory.
`small-file-space-in-bytes`	`1048576`	(optional) Specifies the total size of all files that are read into memory.
`file-transmission-enabled`	`false`	(optional) If `true`, enables the use of `TransmitFileSystem` calls. Meaningful only for Windows.
`max-files-count`	`1024`	(optional) Specifies the maximum number of files in the file cache.
`hash-init-size`	`0`	(optional) Specifies the initial number of hash buckets.

http-listener

Defines an HTTP listen socket. The "connection-pool" on page 40 subelement of the parent "http-service" on page 79 element also configures some listen socket settings.

Superelements

"http-service" on page 79

Subelements

The following table describes subelements for the `http-listener` element.

TABLE 1–79 `http-listener` Subelements

Element	Required	Description
"ssl" on page 150	zero or one	Defines Secure Socket Layer (SSL) parameters.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `http-listener` element.

TABLE 1–80 `http-listener` Attributes

Attribute	Default	Description
`id`	none	The unique listener name. An `http-listener` name cannot begin with a number.
`address`	none	IP address of the listener. Can be in dotted-pair or IPv6 notation. Can be `any` (for `INADDR_ANY`) to listen on all IP addresses. Can be a hostname.
`port`	none	Port number on which the listener listens. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges. Configuring an SSL listener to listen on port 443 is standard.
`external-port`	none	(optional) Specifies the external port on which the connection is made.
`family`		(optional) Deprecated. Do not use.
`blocking-enabled`	`false`	(optional) If `true`, uses a blocking socket for servicing a request.
`acceptor-threads`	`1`	(optional) Specifies the number of processors in the machine. To set the number of request processing threads, use the `thread-count` attribute of the "request-processing" on page 131 element.
`security-enabled`	`false`	(optional) Determines whether the listener runs SSL. To turn SSL2 or SSL3 on or off and set ciphers, use an `ssl` subelement.
`default-virtual-server`	none	References the `id` attribute of the default "virtual-server" on page 162 for this particular listener.

TABLE 1–80 `http-listener` Attributes *(Continued)*

Attribute	Default	Description
`server-name`	none	Tells the server what to put in the host name section of any URLs it sends to the client. This affects URLs the server automatically generates; it doesn't affect the URLs for directories and files stored in the server. If your server uses an alias, the `server-name` should be the alias name. If a colon and port number are appended, that port is used in URLs the server sends to the client. If load balancing is enabled, use the server name of the load balancer.
`redirect-port`	none	(optional) If the listener is supporting non-SSL requests and a request is received for which a matching `<security-constraint>` requires SSL transport, the request is automatically redirected to the port number specified here. If load balancing is enabled, use the redirect port of the load balancer.
`xpowered-by`	`true`	(optional) If true, `X-Powered-By` headers are used according to the Servlet 2.4 and JSP 2.0 specifications.
`enabled`	`true`	(optional) Determines whether the listener is active. If set to `false`, any attempts to connect to the listener result in a socket exception (`java.net.ConnectException`). In Communications Application Server versions prior to 9.1, a listener whose `enabled` attribute was set to `false` returned a 404 response code for any requests sent to it. To achieve this behavior in the current Communications Application Server version, set the listener's `enabled` attribute to `true`, and set every associated virtual server's state to `off`. A "virtual-server" on page 162 lists its associated listeners in its `http-listeners` attribute.

Properties

The following table describes properties for the `http-listener` element. Any of these properties can be defined as an "http-service" on page 79 property, so that it applies to all `http-listener` elements.

TABLE 1–81 `http-listener` Properties

Property	Default	Description
`recycle-objects`	`true`	If `true`, recycles internal objects instead of using the VM garbage collector.
`reader-threads`	`0`	Specifies the number of reader threads, which read bytes from the non-blocking socket.
`acceptor-queue-length`	`4096`	Specifies the length of the acceptor thread queue. Once full, connections are rejected.
`reader-queue-length`	`4096`	Specifies the length of the reader thread queue. Once full, connections are rejected.
`use-nio-direct-bytebuffer`	`true`	If `true`, specifies that the NIO direct `ByteBuffer` is used. In a limited resource environment, it might be faster to use non-direct Java's `ByteBuffer` by setting a value of `false`.

TABLE 1–81 `http-listener` Properties *(Continued)*

Property	Default	Description
`authPassthrough Enabled`	`false`	If `true`, indicates that this `http-listener` element receives traffic from an SSL-terminating proxy server. Overrides the `authPassthroughEnabled` property of the parent "http-service" on page 79 element.
`proxyHandler`	`com.sun.enterprise. web.ProxyHandlerImpl`	Specifies the fully qualified class name of a custom implementation of the `com.sun.appserv.ProxyHandler` abstract class that this `http-listener` uses. Only used if the `authPassthroughEnabled` property of this `http-listener` and the parent "http-service" on page 79 element are both set to `true`. Overrides the `proxyHandler` property of the parent `http-service` element.
`proxiedProtocol`	none	Specifies a comma-separated list of protocols that can use the same port. Allowed values are `ws/tcp` (SOAP over TCP), `http`, `https` and `tls`. For example, if you set this property to `http,https` and set the `port` to `4567`, you can access the port with either `http://`*host*`:4567/` or `https://`*host*`:4567/`. Specifying this property at the "http-service" on page 79 level overrides settings at the `http-listener` level. If this property is not set at either level, this feature is disabled.
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`30`	Specifies the number of seconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.
`maxKeepAliveRequests`	`250`	Specifies the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Set this property to `1` to disable HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining.
`traceEnabled`	`true`	If `true`, enables the TRACE operation. Set this property to `false` to make the Communications Application Server less susceptible to cross-site scripting attacks.

`cometSupport`	`false`	If `true`, enables Comet support for this listener.
		If your servlet or JSP page uses Comet technology, make sure it is initialized when the Communications Application Server starts up by adding the `load-on-startup` element to your `web.xml` file. For example:
		`<servlet>`
		`  <servlet-name>CheckIn</servlet-name>`
		`  <servlet-class>CheckInServlet</servlet-class>`
		`  <load-on-startup>0</load-on-startup>`
		`</servlet>`

TABLE 1–81 `http-listener` Properties *(Continued)*

Property	Default	Description
`compression`	`off`	Specifies use of HTTP/1.1 GZIP compression to save server bandwidth. Allowed values are: ■ `off` – Disables compression. ■ `on` – Compresses data. ■ `force` – Forces data compression in all cases. ■ positive integer – Specifies the minimum amount of data required before the output is compressed. If the `content-length` is not known, the output is compressed only if compression is set to `on` or `force`.
`compressableMimeType`	`text/html,text/xml, text/plain`	Specifies a comma-separated list of MIME types for which HTTP compression is used.
`noCompression UserAgents`	empty `String` (`regexp` matching disabled)	Specifies a comma-separated list of regular expressions matching user-agents of HTTP clients for which compression should not be used.
`minCompressionSize`	none	Specifies the minimum size of a file when compression is applied.
`crlFile`	none	Specifies the location of the Certificate Revocation List (CRL) file to consult during SSL client authentication. This can be an absolute or relative file path. If relative, it is resolved against *domain-dir*. If unspecified, CRL checking is disabled.
`trustAlgorithm`	none	Specifies the name of the trust management algorithm (for example, `PKIX`) to use for certification path validation.
`trustMaxCertLength`	`5`	Specifies the maximum number of non-self-issued intermediate certificates that can exist in a certification path. This property is considered only if `trustAlgorithm` is set to `PKIX`. A value of zero implies that the path can only contain a single certificate. A value of `-1` implies that the path length is unconstrained (there is no maximum). Setting a value less than `-1` causes an exception to be thrown.
`disableUploadTimeout`	`true`	if `false`, the connection for a servlet that reads bytes slowly is closed after the `connectionUploadTimeout` is reached.
`connectionUpload Timeout`	`5`	Specifies the timeout for uploads. Applicable only if `disableUploadTimeout` is set to `false`.
`uriEncoding`	`UTF-8`	Specifies the character set used to decode the request URIs received on this HTTP listener. Must be a valid IANA character set name. Overrides the `uriEncoding` property of the parent "http-service" on page 79 element.

http-protocol

Configures HTTP protocol settings.

Superelements

"http-service" on page 79

Subelements

none

Attributes

The following table describes attributes for the `http-protocol` element.

TABLE 1–82 `http-protocol` Attributes

Attribute	Default	Description
`version`	`HTTP/1.1`	(optional) Specifies the version of the HTTP protocol used.
`dns-lookup-enabled`	`true`	(optional) If `true`, looks up the DNS entry for the client.
`forced-type`	`ISO-8859-1;en;ISO-8859-1`	(optional) Specifies the request type used if no MIME mapping is available that matches the file extension. The format is a semicolon-delimited string consisting of the content-type, encoding, language, and charset.
`default-type`	`text/html;ISO-8859-1;en;ISO-8859-1`	(optional) Specifies the default response type. The format is a semicolon-delimited string consisting of the content-type, encoding, language, and charset.
`forced-response-type`	`AttributeDeprecated`	(optional) Deprecated. Do not use.
`default-response-type`	`AttributeDeprecated`	(optional) Deprecated. Do not use.
`ssl-enabled`	`true`	(optional) Not implemented. Use `ssl` subelements of "http-listener" on page 74 elements.

http-service

Defines the HTTP service.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `http-service` element.

TABLE 1–83 `http-service` Subelements

Element	Required	Description
"access-log" on page 22	zero or one	Defines access log settings for each "http-access-log" on page 73 subelement of each "virtual-server" on page 162.
"http-listener" on page 74	one or more	Defines an HTTP listen socket.
"virtual-server" on page 162	one or more	Defines a virtual server.
"request-processing" on page 131	zero or one	Configures request processing threads.
"keep-alive" on page 100	zero or one	Configures keep-alive threads.
"connection-pool" on page 40	zero or one	Defines a pool of client HTTP connections.
"http-protocol" on page 78	zero or one	Configures HTTP protocol settings.
"property" on page 125	zero or more	Specifies a property or a variable.

Properties

The following table describes properties for the `http-service` element. These properties apply to all "http-listener" on page 74 subelements, except for `accessLoggingEnabled`, `accessLogBufferSize`, and `accessLogWriterInterval`, which apply to all "virtual-server" on page 162 subelements.

TABLE 1–84 `http-service` Properties

Property	Default	Description
`monitoring-cache-enabled`	`true`	If `true`, enables the monitoring cache.
`monitoring-cache-refresh-in-millis`	`5000`	Specifies the interval between refreshes of the monitoring cache.
`ssl-cache-entries`	`10000`	Specifies the number of SSL sessions to be cached.
`ssl3-session-timeout`	`86400`	Specifies the interval at which SSL3 sessions are cached.
`ssl-session-timeout`	`100`	Specifies the interval at which SSL2 sessions are cached.
`recycle-objects`	`true`	If `true`, recycles internal objects instead of using the VM garbage collector.
`reader-threads`	`0`	Specifies the number of reader threads, which read bytes from the non-blocking socket.
`acceptor-queue-length`	`4096`	Specifies the length of the acceptor thread queue. Once full, connections are rejected.
`reader-queue-length`	`4096`	Specifies the length of the reader thread queue. Once full, connections are rejected.

TABLE 1–84 `http-service` Properties *(Continued)*

Property	Default	Description
`use-nio-direct-bytebuffer`	`true`	If `true`, specifies that the NIO direct `ByteBuffer` is used. In a limited resource environment, it might be faster to use non-direct Java's `ByteBuffer` by setting a value of `false`.
`authPassthroughEnabled`	`false`	If `true`, indicates that the "http-listener" on page 74 subelements receive traffic from an SSL-terminating proxy server, which is responsible for forwarding any information about the original client request (such as client IP address, SSL `keysize`, and authenticated client certificate chain) to the HTTP listeners using custom request headers. Each `http-listener` subelement can override this setting for itself.
`proxyHandler`	`com.sun.enterprise.web.ProxyHandlerImpl`	Specifies the fully qualified class name of a custom implementation of the `com.sun.appserv.ProxyHandler` abstract class, which allows a back-end application server instance to retrieve information about the original client request that was intercepted by an SSL-terminating proxy server (for example, a load balancer). An implementation of this abstract class inspects a given request for the custom request headers through which the proxy server communicates the information about the original client request to the Communications Application Server instance, and returns that information to its caller. The default implementation reads the client IP address from an HTTP request header named `Proxy-ip`, the SSL `keysize` from an HTTP request header named `Proxy-keysize`, and the SSL client certificate chain from an HTTP request header named `Proxy-auth-cert`. The `Proxy-auth-cert` value must contain the BASE-64 encoded client certificate chain without the `BEGIN CERTIFICATE` and `END CERTIFICATE` boundaries and with `\n` replaced with `% d% a`. Only used if `authPassthroughEnabled` is set to `true`. Each "http-listener" on page 74 subelement can override the `proxyHandler` setting for itself.
`proxiedProtocol`	none	Specifies a comma-separated list of protocols that can use the same port. Allowed values are `ws/tcp` (SOAP over TCP), `http`, `https` and `tls`. For example, if you set this property to `http,https` and the `port` is `4567`, you can access the port with either `http://`*host*`:4567/` or `https://`*host*`:4567/`. Specifying this property at the `http-service` level overrides settings at the "http-listener" on page 74 level. If this property is not set at either level, this feature is disabled.
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`30`	Specifies the number of seconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.

TABLE 1–84 `http-service` Properties *(Continued)*

Property	Default	Description
`maxKeepAliveRequests`	`250`	Specifies the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Set this property to `1` to disable HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining.
`traceEnabled`	`true`	If `true`, enables the TRACE operation. Set this property to `false` to make the Communications Application Server less susceptible to cross-site scripting attacks.
`accessLoggingEnabled`	`false` (developer and cluster profiles) `true` (enterprise profile)	If `true`, enables access logging for all "virtual-server" on page 162 subelements that do not specify this property. If `false`, disables access logging for all `virtual-server` subelements that do not specify this property.
`accessLogBufferSize`	`32768`	Specifies the size, in bytes, of the buffer where access log calls are stored. If the value is less than `5120`, a warning message is issued, and the value is set to `5120`.
`accessLogWriterInterval`	`300`	Specifies the number of seconds before the log is written to the disk. The access log is written when the buffer is full or when the interval expires. If the value is `0`, the buffer is always written even if it is not full. This means that each time the server is accessed, the log message is stored directly to the file.
`sso-enabled`	`false` (developer and cluster profiles) `true` (enterprise profile)	If `true`, single sign-on is enabled by default for all web applications on all virtual servers on this server instance that are configured for the same realm. If `false`, single sign-on is disabled by default for all virtual servers, and users must authenticate separately to every application on each virtual server. The `sso-enabled` property setting of the "virtual-server" on page 162 element overrides this setting for an individual virtual server. At the `http-service` level, you cannot change the `sso-max-inactive-seconds` and `sso-reap-interval-seconds` values from their defaults. However, you can change these values at the `virtual-server` level.
`disableUploadTimeout`	`true`	if `false`, the connection for a servlet that reads bytes slowly is closed after the `connectionUploadTimeout` is reached.
`connectionUploadTimeout`	`5`	Specifies the timeout for uploads. Applicable only if `disableUploadTimeout` is set to `false`.
`uriEncoding`	`UTF-8`	Specifies the character set used to decode the request URIs received on "http-listener" on page 74 subelements that do not define this property. Must be a valid IANA character set name.

I

identity-assertion-trust

Specifies identity assertion trust domain configuration information according to RFC 3325. P-asserted identity headers received from hosts and domains configured under this element can be trusted. A P-asserted identity header specifies the identity of a user who was authenticated at another node in the network.

If no `trusted-entity` or `trust-handler` subelements are defined, this represents presumed trust, that is, any identity assertion that is received is trusted based on a presumption that the network topology would prevent untrusted assertions from reaching the server. This is distinguished from the case where no `identity-assertion-trust` element has been configured, in which case no identity assertions are to be accepted.

Superelements

"security-service" on page 136

Subelements

The following table describes subelements for the `identity-assertion-trust` element.

TABLE 1–85 `identity-assertion-trust` Subelements

Element	Required	Description
"trusted-entity" on page 160	zero or more if no `trust-handler` is defined	Specifies intermediate hosts and domains according to RFC 3325.
"trust-handler" on page 161	zero or one if no `trusted-entity` is defined	Specifies a custom trust handler according to RFC 3325.

Attributes

The following table describes attributes for the `identity-assertion-trust` element.

TABLE 1–86 `identity-assertion-trust` Attributes

Attribute	Default	Description
`id`	none	Specifies a unique identifier for the `identity-assertion-trust` element.

TABLE 1–86 identity-assertion-trust Attributes (Continued)

Attribute	Default	Description
is-default	false	If true, specifies that this is the default identity-assertion-trust. There can be only one default identity-assertion-trust.

iiop-listener

Defines an IIOP listen socket. To enable SSL for this listener, include an ssl subelement.

Superelements

"iiop-service" on page 84

Subelements

The following table describes subelements for the iiop-listener element.

TABLE 1–87 iiop-listener Subelements

Element	Required	Description
"ssl" on page 150	zero or one	Defines SSL parameters.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the iiop-listener element.

TABLE 1–88 iiop-listener Attributes

Attribute	Default	Description
id	none	The listener name. An iiop-listener name cannot begin with a number.
address	none	IP address of the listener. Can be in dotted-pair or IPv6 notation, or just a name.
port	1072	(optional) Port number for the listener. Legal values are 1 - 65535. On UNIX, creating sockets that listen on ports 1 - 1024 requires superuser privileges.
security-enabled	false	(optional) Determines whether the listener runs SSL. To turn SSL2 or SSL3 on or off and set ciphers, use an ssl element.
enabled	true	(optional) Determines whether the listener is active.

iiop-service

Defines the IIOP service.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `iiop-service` element.

TABLE 1–89 `iiop-service` Subelements

Element	Required	Description
"orb" on page 122	only one	Configures the ORB.
"ssl-client-config" on page 152	zero or one	Defines SSL parameters for the ORB.
"iiop-listener" on page 84	zero or more	Defines an IIOP listen socket.

Attributes

The following table describes attributes for the `iiop-service` element.

TABLE 1–90 `iiop-service` Attributes

Attribute	Default	Description
`client-authentication-required`	`false`	(optional) If `true`, the server rejects unauthenticated requests and inserts an authentication-required bit in IORs sent to clients.

ip-address

Identifies the trusted host on the network using an IP address.

Superelements

"trusted-entity" on page 160

Subelements

none - contains data

J

j2ee-application

Specifies a deployed Java EE application.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `j2ee-application` element.

TABLE 1–91 `j2ee-application` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 172	zero or more	Configures a web service endpoint.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `j2ee-application` element.

TABLE 1–92 `j2ee-application` Attributes

Attribute	Default	Description
`name`	none	The name of the application.
`location`	none	The location of the application in the Communications Application Server file system.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the application is enabled.

TABLE 1–92 `j2ee-application` Attributes *(Continued)*

Attribute	Default	Description
`libraries`	none	(optional) Specifies an absolute or relative path to libraries specific to this module or application. A relative path is relative to *domain-dir*`/lib/applibs`. If the path is absolute, the path must be accessible to the domain administration server (DAS), which means it must be under *domain-dir*. To include more than one path, use a system-specific separator, such as a colon for Solaris or a semicolon for Windows. The libraries are made available to the application in the order in which they are specified.
`availability-enabled`	`false`	(optional) Specifies whether availability is enabled in this Java EE application for HTTP session persistence and SFSB checkpointing (and potentially passivation). Availability must also be enabled for the application during deployment. For more information about availability, see "availability-service" on page 33.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.
`java-web-start-enabled`	`true`	(optional) Specifies whether Java Web Start access is permitted for application clients in this application.

jacc-provider

Specifies a Java Authorization Contract for Containers (JACC) provider for pluggable authorization.

Superelements

"security-service" on page 136

Subelements

The following table describes subelements for the `jacc-provider` element.

TABLE 1–93 `jacc-provider` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jacc-provider` element.

TABLE 1–94 `jacc-provider` Attributes

Attribute	Default	Description
`name`	`default`	Specifies the name of the JACC provider.
`policy-provider`	`com.sun.enterprise.security.provider.PolicyWrapper`	Corresponds to and can be overridden by the system property `javax.security.jacc.policy.provider`.
`policy-configuration-factory-provider`	`com.sun.enterprise.security.provider.PolicyConfigurationFactoryImpl`	Corresponds to and can be overridden by the system property `javax.security.jacc.PolicyConfigurationFactory.provider`.

java-config

Specifies Java Virtual Machine (JVM) configuration parameters.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `java-config` element.

TABLE 1–95 `java-config` Subelements

Element	Required	Description
"profiler" on page 124	zero or one	Configures a profiler for use with the Communications Application Server.
"jvm-options" on page 100	zero or more	Contains JVM command line options.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `java-config` element.

TABLE 1–96 `java-config` Attributes

Attribute	Default	Description
`java-home`	none	The path to the directory where the JDK is installed.
`debug-enabled`	`false`	(optional) If `true`, the server starts up in debug mode ready for attachment with a JPDA-based debugger.

TABLE 1–96 `java-config` Attributes *(Continued)*

Attribute	Default	Description
`debug-options`	`-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n`	(optional) Specifies JPDA (Java Platform Debugger Architecture) options. A list of debugging options is available at http://java.sun.com/products/jpda/doc/conninv.html#Invocation. For more information about debugging, see the *Sun Java System Application Server 9.1 Developer's Guide*.
`rmic-options`	`-iiop -poa -alwaysgenerate -keepgenerated -g`	(optional) Specifies options passed to the RMI compiler at application deployment time. The `-keepgenerated` option saves generated source for stubs and ties. For details about the `rmic` command, see http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/rmic.html.
`javac-options`	`-g`	(optional) Specifies options passed to the Java compiler at application deployment time.
`classpath-prefix`	none	(optional) Specifies a prefix for the server classpath. Only prefix this classpath to override Communications Application Server classes. Use this attribute with caution.
`classpath-suffix`	none	(optional) Specifies a suffix for the server classpath.
`server-classpath`	none	(optional) Specifies additions to the server classpath. Supported for backward compatibility. Use `classpath-suffix` instead.
`system-classpath`	JVM classes	(optional) Specifies additions to the system classpath, which is supplied to the JVM at server startup. These classes are loaded by the System Classloader. **Note –** Do not remove the default path.
`native-library-path-prefix`	none	(optional) Specifies a prefix for the native library path. The native library path is the automatically constructed concatenation of the Communications Application Server installation relative path for its native shared libraries, the standard JRE native library path, the shell environment setting (`LD_LIBRARY_PATH` on UNIX), and any path specified in the `profiler` element. Since this is synthesized, it does not appear explicitly in the server configuration.
`native-library-path-suffix`	none	(optional) Specifies a suffix for the native library path.
`bytecode-preprocessors`	none	(optional) A comma separated list of class names, each of which must implement the `com.sun.appserv.BytecodePreprocessor` interface. Each of the specified preprocessor classes is called in the order specified.
`env-classpath-ignored`	`true`	(optional) If `false`, the `CLASSPATH` environment variable is read and appended to the Communications Application Server classpath. The `CLASSPATH` environment variable is added after the `classpath-suffix`, at the very end. For a development environment, this value should be set to `false`. To prevent environment variable side effects in a production environment, set this value to `true`.

jdbc-connection-pool

Defines the properties that are required for creating a JDBC connection pool.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `jdbc-connection-pool` element.

TABLE 1–97 `jdbc-connection-pool` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jdbc-connection-pool` element. Changing the following attributes requires a server restart: `datasource-classname`, `associate-with-thread`, `lazy-connection-association`, and `lazy-connection-enlistment`.

TABLE 1–98 `jdbc-connection-pool` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connection pool. A "jdbc-resource" on page 94 element's `pool-name` attribute refers to this `name`.
`datasource-classname`	none	Specifies the class name of the associated vendor-supplied data source. This class must implement `javax.sql.DataSource`, `javax.sql.XADataSource` , `javax.sql.ConnectionPoolDatasource`, or a combination.
`res-type`	`javax.sql.DataSource`	(optional) Specifies the interface the data source class implements. The value of this attribute can be `javax.sql.DataSource`, `javax.sql.XADataSource` , or `javax.sql.ConnectionPoolDatasource`. If the value is not one of these interfaces, the default is used. An error occurs if this attribute has a legal value and the indicated interface is not implemented by the data source class.
`steady-pool-size`	8	(optional) Specifies the initial and minimum number of connections maintained in the pool.
`max-pool-size`	32	(optional) Specifies the maximum number of connections that can be created to satisfy client requests.

TABLE 1–98 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`max-wait-time-in-millis`	`60000`	(optional) Specifies the amount of time, in milliseconds, that the caller is willing to wait for a connection. If `0`, the caller is blocked indefinitely until a resource is available or an error occurs.
`pool-resize-quantity`	`2`	(optional) Specifies the number of idle connections to be destroyed if the existing number of connections is above the `steady-pool-size` (subject to the `max-pool-size` limit). This is enforced periodically at the `idle-timeout-in-seconds` interval. An idle connection is one that has not been used for a period of `idle-timeout-in-seconds`. When the pool size reaches `steady-pool-size`, connection removal stops.
`idle-timeout-in-seconds`	`300`	(optional) Specifies the maximum time that a connection can remain idle in the pool. After this amount of time, the pool can close this connection.
`transaction-isolation-level`	default JDBC driver isolation level	(optional) Specifies the transaction isolation level on the pooled database connections. Allowed values are `read-uncommitted`, `read-committed` , `repeatable-read`, or `serializable`. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool, which can lead to errors. See `is-isolation-level-guaranteed` for more details.
`is-isolation-level-guaranteed`	`true`	(optional) Applicable only when `transaction-isolation-level` is explicitly set. If `true`, every connection obtained from the pool is guaranteed to have the desired isolation level. This might impact performance on some JDBC drivers. Only set this attribute to `false` if you are certain that the hosted applications do not return connections with altered isolation levels.
`is-connection-validation-required`	`false`	(optional) Specifies whether connections have to be validated before being given to the application. If a resource's validation fails, it is destroyed, and a new resource is created and returned.
`connection-validation-method`	`auto-commit`	(optional) Legal values are as follows: ■ `auto-commit` (default), which uses `Connection.setAutoCommit(Connection.getAutoCommit())` ■ `meta-data`, which uses `Connection.getMetaData()` ■ `table`, which performs a query on a table specified in the `validation-table-name` attribute
`validation-table-name`	none	(optional) Specifies the table name to be used to perform a query to validate a connection. This parameter is mandatory if and only if `connection-validation-method` is set to `table`.
`fail-all-connections`	`false`	(optional) If `true`, closes all connections in the pool if a single validation check fails. This parameter is mandatory if and only if `is-connection-validation-required` is set to `true`.
`non-transactional-connections`	`false`	(optional) If `true`, non-transactional connections can be made to the JDBC connection pool. These connections are not automatically enlisted with the transaction manager.

TABLE 1–98 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`allow-non-component-callers`	`false`	(optional) If `true`, non-Java-EE components, such as servlet filters, lifecycle modules, and third party persistence managers, can use this JDBC connection pool. The returned connection is automatically enlisted with the transaction context obtained from the transaction manager. Standard Java EE components can also use such pools. Connections obtained by non-component callers are not automatically closed at the end of a transaction by the container. They must be explicitly closed by the caller.
`connection-leak-timeout-in-seconds`	`0`	Detects potential connection leaks by the application. A connection that is not returned back to the pool by the application within the specified period is assumed to be potentially leaking, and a stack trace of the caller is logged. A zero value disables leak detection. A nonzero value enables leak tracing.
`connection-leak-reclaim`	`false`	If `true`, the pool will reclaim a connection after `connection-leak-timeout-in-seconds` occurs.
`connection-creation-retry-attempts`	`0`	Specifies the number of attempts to create a new connection.
`connection-creation-retry-interval-in-seconds`	`10`	Specifies the time interval between attempts to create a connection when `connection-creation-retry-attempts` is greater than `0`.
`validate-atmost-once-period-in-seconds`	`0`	Specifies the time interval within which a connection is validated at most once. Minimizes the number of validation calls.
`statement-timeout-in-seconds`	`-1`	Sets the query timeout property of a statement to enable termination of abnormally long running queries. The default value of `-1` disables this feature.
`lazy-connection-enlistment`	`false`	If `true`, a connection is not enlisted in a transaction until it is used. If `false`, any connection object available to a transaction is enlisted in the transaction.
`lazy-connection-association`	`false`	If `true`, a physical connection is not associated with a logical connection until it is used. If `false`, a physical connection is associated with a logical connection even before it is used.
`associate-with-thread`	`false`	If `true`, allows a connection to be saved as a `ThreadLocal` in the calling thread. This connection gets reclaimed only when the calling thread dies or when the calling thread is not in use and the pool has run out of connections.
`match-connections`	`false`	If `true`, enables connection matching. You can set to `false` if connections are homogeneous.
`max-connection-usage-count`	`0`	Specifies the number of times a connections is reused by the pool, after which it is closed. A zero value disables this feature.
`wrap-jdbc-objects`	`false`	If `true`, the application returns wrapped JDBC objects for `Statement`, `PreparedStatement`, `CallableStatement`, `ResultSet`, and `DatabaseMetaData`.

Properties

Most JDBC drivers allow use of standard property lists to specify the user, password, and other resource configuration information. Although properties are optional with respect to the Communications Application Server, some properties might be necessary for most databases. For details, see the JDBC 4.0 Standard Extension API.

When properties are specified, they are passed to the vendor's data source class (specified by the `datasource-classname` attribute) as is using set*Name*(*value*) methods.

The `user` and `password` properties are used as the default principal if container managed authentication is specified and a `default-resource-principal` is not found in the application deployment descriptors.

The following table describes some common properties for the `jdbc-connection-pool` element.

Changing JDBC driver properties requires a server restart.

TABLE 1–99 `jdbc-connection-pool` Properties

Property	Description
`user`	Specifies the user name for connecting to the database.
`password`	Specifies the password for connecting to the database.
`databaseName`	Specifies the database for this connection pool.
`serverName`	Specifies the database server for this connection pool.
`port`	Specifies the port on which the database server listens for requests.
`networkProtocol`	Specifies the communication protocol.
`roleName`	Specifies the initial SQL role name.
`datasourceName`	Specifies an underlying `XADataSource`, or a `ConnectionPoolDataSource` if connection pooling is done.
`description`	Specifies a text description.
`url`	Specifies the URL for this connection pool. Although this is not a standard property, it is commonly used.
`LazyConnection Enlistment`	Deprecated. Use the equivalent attribute.
`LazyConnection Association`	Deprecated. Use the equivalent attribute.
`AssociateWithThread`	Deprecated. Use the equivalent attribute.
`MatchConnections`	Deprecated. Use the equivalent attribute.

jdbc-resource

Defines a JDBC (`javax.sql.DataSource`) resource.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `jdbc-resource` element.

TABLE 1–100 `jdbc-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jdbc-resource` element.

TABLE 1–101 `jdbc-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`pool-name`	none	Specifies the `name` of the associated "jdbc-connection-pool" on page 90.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

jms-availability

Enables availability in the Sun Java System Message Queue cluster that comprises the Java Message Service (JMS). Messages are saved to the HADB. The HADB must be installed and the enterprise profile must be selected. You must enable availability for Communications Application Server instances before you can enable availability for the corresponding Message Queue brokers.

Note – Individual applications and modules cannot control or override JMS availability.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"availability-service" on page 33

Subelements

The following table describes subelements for the `jms-availability` element.

TABLE 1–102 `jms-availability` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jms-availability` element.

TABLE 1–103 `jms-availability` Attributes

Attribute	Default	Description
`availability -enabled`	`false`	(optional) If set to `true`, and if availability is enabled for the Communications Application Server instance (see "availability-service" on page 33), high-availability is enabled for the Message Queue cluster associated with the Communications Application Server cluster. All instances in an Communications Application Server cluster should have the same availability settings to ensure consistent behavior.

jms-host

Configures the host of the built-in Java Message Service (JMS) that is managed by the Communications Application Server.

Superelements

"jms-service" on page 96

Subelements

The following table describes subelements for the `jms-host` element.

TABLE 1–104 `jms-host` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jms-host` element.

TABLE 1–105 `jms-host` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the JMS host.
`host`	*machine-name*	(optional) Specifies the host name of the JMS host.
`port`	`7676`	(optional) Specifies the port number used by the JMS provider.
`admin-user-name`	`admin`	(optional) Specifies the administrator user name for the JMS provider.
`admin-password`	`admin`	(optional) Specifies the administrator password for the JMS provider.

jms-service

Configures the built-in Java Message Service (JMS) that is managed by the Communications Application Server.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `jms-service` element.

TABLE 1–106 `jms-service` Subelements

Element	Required	Description
"jms-host" on page 95	zero or more	Specifies a host.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jms-service` element.

TABLE 1–107 `jms-service` Attributes

Attribute	Default	Description
`init-timeout-in-seconds`	`60`	(optional) Specifies the amount of time the server instance waits at startup for its configured default JMS host to respond. If there is no response, startup is aborted. If set to `0`, the server instance waits indefinitely.
`type`	`EMBEDDED` (DAS) or `LOCAL` (other server instances)	Specifies the type of JMS service: ■ `EMBEDDED` means the JMS provider is started in the same JVM as the Communications Application Server, and the networking stack is bypassed. Lazy initialization starts the default embedded broker on the first access of JMS services rather than at Communications Application Server startup. `EMBEDDED` mode is not a supported configuration for a cluster. ■ `LOCAL` means the JMS provider is started along with the Communications Application Server. The `LOCAL` setting implicitly sets up a 1:1 relationship between an Communications Application Server instance and a Message Queue broker. When you create an Communications Application Server cluster, a Message Queue cluster is automatically created as well. During cluster creation, each instance in the Communications Application Server cluster is automatically configured with a broker in the Message Queue cluster, and a unique broker port is determined. ■ `REMOTE` means the JMS provider is remote and is not started by the Communications Application Server.
`start-args`	none	(optional) Specifies the string of arguments supplied for startup of the corresponding JMS instance.
`default-jms-host`	none	Specifies the name of the default "jms-host" on page 95. If `type` is set to `LOCAL`, this `jms-host` is automatically started at Communications Application Server startup.
`reconnect-interval-in-seconds`	`5` (developer profile) `60` (cluster and enterprise profiles)	(optional) Specifies the interval between reconnect attempts.
`reconnect-attempts`	`3`	(optional) Specifies the number of reconnect attempts.

TABLE 1–107 `jms-service` Attributes *(Continued)*

Attribute	Default	Description
`reconnect-enabled`	`true`	(optional) If `true`, reconnection is enabled. The JMS service automatically tries to reconnect to the JMS provider when the connection is broken. When the connection is broken, depending on the message processing stage, the `onMessage()` method might not be able to complete successfully or the transaction might be rolled back due to a JMS exception. When the JMS service reestablishes the connection, JMS message redelivery semantics apply.
`addresslist-behavior`	`random`	(optional) Specifies whether the reconnection logic selects the broker from the `imqAddressList` in a `random` or sequential (`priority`) fashion.
`addresslist-iterations`	`3`	(optional) Specifies the number of times the reconnection logic iterates over the `imqAddressList` if `addresslist-behavior` is set to `PRIORITY`.
`mq-scheme`	`mq`	(optional) Specifies the scheme for establishing connection with the broker. For example, specify `http` for connecting to the broker over HTTP.
`mq-service`	`jms`	(optional) Specifies the type of broker service. If a broker supports SSL, the type of service can be `ssljms`.

Properties

The following table describes properties for the `jms-service` element.

TABLE 1–108 `jms-service` Properties

Property	Default	Description
`instance-name`	`imqbroker`	Specifies the full Sun Java System Message Queue broker instance name.
`instance-name-suffix`	none	Specifies a suffix to add to the full Message Queue broker instance name. The suffix is separated from the instance name by an underscore character (_). For example, if the instance name is `imqbroker`, appending the suffix `xyz` changes the instance name to `imqbroker_xyz`.
`append-version`	`false`	If `true`, appends the major and minor version numbers, preceded by underscore characters (_), to the full Message Queue broker instance name. For example, if the instance name is `imqbroker` , appending the version numbers changes the instance name to `imqbroker_8_0`.
`user-name`	`guest`	Specifies the user name for creating the JMS connection. Needed only if the default username/password of guest/guest is not available in the broker.
`password`	`guest`	Specifies the password for creating the JMS connection. Needed only if the default username/password of guest/guest is not available in the broker.

jmx-connector

Configures a JSR 160/255 compliant remote JMX connector, which handles the JMX communication between the domain administration server, the node agents, and the remote

server instances. This JMX connector also handles JMX communication between an external management client and the domain administration server.

Only the system JMX connector is started by the server processes at startup. Do not configure additional JMX connectors.

Superelements

"admin-service" on page 25, "node-agent" on page 120

Subelements

The following table describes subelements for the `jmx-connector` element.

TABLE 1–109 `jmx-connector` Subelements

Element	Required	Description
"ssl" on page 150	zero or one	Defines SSL parameters.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `jmx-connector` element.

TABLE 1–110 `jmx-connector` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connector used by the designated system JMX connector for JMX communication between server instances. Do not modify this name.
`protocol`	`rmi_jrmp`	(optional) Specifies the protocol that this JMX connector supports. The only supported protocol is `rmi_jrmp`. Do not modify this value.
`address`	`0.0.0.0`	Specifies the IP address of the naming service where the JMX connector server stub is registered. This is not the port of the server socket that does the actual JMX communication. This is the address of the network interface where the RMI registry is started. If your system has multiple network interfaces, modify this value so that only a particular interface is selected.

TABLE 1–110 `jmx-connector` Attributes *(Continued)*

Attribute	Default	Description
`port`	`8686` (DAS, all profiles; server instance, developer profile) `38686` (server instances, cluster and enterprise profiles)	Specifies the port number on with the naming service (RMI registry) listens for RMI client connections. The only use of this naming service is to download the RMI stubs. If the default port is occupied, a free port is used. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges.
`auth-realm-name`	`admin-realm`	Specifies the `name` of an "auth-realm" on page 30 subelement of the "security-service" on page 136 element for the server instance that is running this JMX connector's server end. Note that this is a dedicated administration security realm.
`security-enabled`	`false` (developer profile) `true` (cluster and enterprise profiles)	(optional) Determines whether JMX communication is encrypted.

jvm-options

Contains JVM command line options, for example:

```
<jvm-options>-Xdebug -Xmx128m</jvm-options>
```

For information about JVM options, see `http://java.sun.com/docs/hotspot/VMOptions.html`.

Superelements

"java-config" on page 88, "profiler" on page 124

Subelements

none - contains data

K

keep-alive

Configures keep-alive threads.

Superelements

"http-service" on page 79, "sip-service" on page 149

Subelements

none

Attributes

The following table describes attributes for the `keep-alive` element.

TABLE 1–111 `keep-alive` Attributes

Attribute	Default	Description
`thread-count`	`1`	(optional) Specifies the number of keep-alive threads. The value must be 1 or greater.
`max-connections`	`256`	(optional) Specifies the maximum number of keep-alive connections. A value of `0` means requests are always rejected. A value of `-1` sets no limit to the number of keep-alive connections.
`timeout-in-seconds`	`60`	(optional) Specifies the maximum time for which a keep alive connection is kept open. A value of `0` or less means keep alive connections are kept open indefinitely.

L

lb-config

Defines a load balancer configuration, which can be referenced by a physical "load-balancer" on page 105.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"lb-configs" on page 102

Subelements

The following table describes subelements for the `lb-config` element.

TABLE 1–112 `lb-config` Subelements

Element	Required	Description
"cluster-ref" on page 37	zero or more; zero if a `server-ref` is defined	References a cluster. This element contains some attributes related to load balancing.
"server-ref" on page 138	zero or more; zero if a `cluster-ref` is defined	References a server instance that does not belong to a cluster. The referenced "server" on page 137 element contains some attributes related to load balancing.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `lb-config` element.

TABLE 1–113 `lb-config` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the load balancer configuration.
`response-timeout-in-seconds`	`60`	(optional) Specifies the time within which a server must return a response or it is considered unhealthy.
`https-routing`	`false`	(optional) If `true`, HTTPS requests to the load balancer result in HTTPS requests to the server. If `false`, HTTPS requests to the load balancer result in HTTP requests to the server.
`reload-poll-interval-in-seconds`	`60`	(optional) Specifies the interval between checks for changes to the load balancer configuration file (`loadbalancer.xml`). When changes are detected, the file is reloaded. A value of zero disables reloading.
`monitoring-enabled`	`false`	(optional) If `true`, enables monitoring of load balancing.

lb-configs

Contains load balancer configurations.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `lb-configs` element.

TABLE 1–114 `lb-configs` Subelements

Element	Required	Description
"lb-config" on page 101	zero or more	Defines a load balancer configuration.

lifecycle-module

Specifies a deployed lifecycle module. For more information about lifecycle modules, see the *Sun Java System Application Server 9.1 Developer's Guide*.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `lifecycle-module` element.

TABLE 1–115 `lifecycle-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `lifecycle-module` element.

TABLE 1–116 `lifecycle-module` Attributes

Attribute	Default	Description
`name`	none	The name of the lifecycle module.
`class-name`	none	The fully qualified name of the lifecycle module's class file, which must implement the `com.sun.appserv.server.LifecycleListener` interface.
`classpath`	value of `application-root` attribute of "domain" on page 56 element	(optional) The classpath for the lifecycle module. Specifies where the module is located.
`load-order`	none	(optional) Determines the order in which lifecycle modules are loaded at startup. Modules with smaller integer values are loaded sooner. Values can range from `101` to the operating system's `MAXINT`. Values from `1` to `100` are reserved.

TABLE 1–116 `lifecycle-module` Attributes *(Continued)*

Attribute	Default	Description
`is-failure-fatal`	`false`	(optional) Determines whether the server is shut down if the lifecycle module fails.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the lifecycle module is enabled.

listener-config

Configures the listener class that listens for alerts from notification emitters. For example:

```
<listener-config
    listener-class-name="com.sun.enterprise.admin.notification.MailAlert"
    subcribe-listener-with="LogMBean,ServerStatusMonitor" >
    <property name="recipients" value="Huey@sun.com,Dewey@sun.com" />
    <property name="fromAddress" value="Louie@sun.com" />
    <property name="subject" value="Help!" />
    <property name="includeDiagnostics" value="false" />
    <property name="mailSMTPHost" value="ducks.sun.com" />
</listener-config>
```

Superelements

"alert-subscription" on page 27

Subelements

The following table describes subelements for the `listener-config` element.

TABLE 1–117 `listener-config` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `listener-config` element.

TABLE 1–118 `listener-config` Attributes

Attribute	Default	Description
`listener-class-name`	none	Specifies the class name of the listener. The `com.sun.appserv.admin.notification.MailAlert` class is provided with the Communications Application Server, but a custom listener can be used.
`subscribe-listener-with`	none	Specifies a comma-separated list of notification emitters to which the listener listens. The `LogMBean` and `ServerStatusMonitor` notification emitters are provided with the Communications Application Server, but custom emitters can be used.

load-balancer

Defines and configures a load balancer. For more information about load balancing in the Communications Application Server, see the *Sun Java System Application Server 9.1 High Availability Administration Guide*.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"load-balancers" on page 106

Subelements

The following table describes subelements for the `load-balancer` element.

TABLE 1–119 `load-balancer` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `load-balancer` element.

TABLE 1–120 `load-balancer` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the load balancer.
`lb-config-name`	none	Specifies the `name` of the "lb-config" on page 101 used by the load balancer.
`auto-apply-enabled`	`false`	(optional) If `true`, changes to the specified "lb-config" on page 101 are automatically applied to the load balancer.

Properties

The following table describes properties for the `load-balancer` element.

TABLE 1–121 `load-balancer` Properties

Property	Default	Description
`device-host`	none	Specifies the host name or IP address for the load balancer.
`device-admin-port`	none	Specifies the load balancer's administration port number.
`ssl-proxy-host`	none	Specifies the load balancer's proxy host used for outbound HTTP.
`ssl-proxy-port`	none	Specifies the load balancer's proxy port used for outbound HTTP.

load-balancers

Contains load balancers.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `load-balancers` element.

TABLE 1–122 `load-balancers` Subelements

Element	Required	Description
"load-balancer" on page 105	zero or more	Defines a load balancer.

log-service

Configures the *server log* file, which stores messages from the default virtual server. Messages from other configured virtual servers also go here, unless the `log-file` attribute is explicitly specified in the `virtual-server` element. The default name is `server.log`.

Other log files are configured by other elements:

- A *virtual server log* file stores messages from a `virtual-server` element that has an explicitly specified `log-file` attribute. See "virtual-server" on page 162.
- The *access log* file stores HTTP access messages from the default virtual server. The default name is `access.log`. See "access-log" on page 22 and "http-access-log" on page 73.
- The *transaction log* files store transaction messages from the default virtual server. The default name of the directory for these files is `tx`. See "transaction-service" on page 157.

Superelements

"config" on page 38, "node-agent" on page 120

Subelements

The following table describes subelements for the `log-service` element.

TABLE 1–123 `log-service` Subelements

Element	Required	Description
"module-log-levels" on page 116	zero or one	Specifies log levels.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `log-service` element.

TABLE 1–124 `log-service` Attributes

Attribute	Default	Description
`file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 56 element	(optional) Overrides the name or location of the server log. The file and directory in which the server log is kept must be writable by the user account under which the server runs. An absolute path overrides the `log-root` attribute of the "domain" on page 56 element. A relative path is relative to the `log-root` attribute of the "domain" on page 56 element. If no `log-root` value is specified, it is relative to *domain-dir*`/config`.
`use-system-logging`	`false`	(optional) If `true`, uses the UNIX `syslog` service to produce and manage logs.
`log-handler`	none	(optional) Specifies a custom log handler to be added to end of the chain of system handlers to log to a different destination.
`log-filter`	none	(optional) Specifies a log filter to do custom filtering of log records.
`log-to-console`	`false`	(optional) Deprecated and ignored.
`log-rotation-limit-in-bytes`	`2000000`	(optional) Log files are rotated when the file size reaches the specified limit.
`log-rotation-timelimit-in-minutes`	`0`	(optional) Enables time-based log rotation. The valid range is `60` minutes (1 hour) to `14400` minutes (10*24*60 minutes or 10 days). If the value is zero, the files are rotated based on the size specified in `log-rotation-limit-in-bytes`. If the value is greater than zero, `log-rotation-timelimit-in-minutes` takes precedence over `log-rotation-limit-in-bytes`.
`retain-error-statistics-for-hours`	`5`	(optional) Specifies the number of most recent hours for which error statistics are retained in memory. The default and minimum value is `5` hours. The maximum value allowed is `500` hours. Larger values incur additional memory overhead.

M

mail-resource

Defines a JavaMail (`javax.mail.Session`) resource.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `mail-resource` element.

TABLE 1–125 `mail-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `mail-resource` element.

TABLE 1–126 `mail-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`store-protocol`	`imap`	(optional) Specifies the storage protocol service, which connects to a mail server, retrieves messages, and saves messages in folder(s). Allowed values are `imap`, `pop3`, `imaps`, and `pop3s` .

`store-protocol-class`	`com.sun.mail.imap.IMAPStore`	(optional) Specifies the service provider implementation class for storage. Allowed values are:
		`com.sun.mail.imap.IMAPStore`
		`com.sun.mail.pop3.POP3Store`
		`com.sun.mail.imap.IMAPSSLStore`
		`com.sun.mail.pop3.POP3SSLStore`

`transport-protocol`	`smtp`	(optional) Specifies the transport protocol service, which sends messages. Allowed values are `smtp` and `smtps`.

`transport-protocol-class`	`com.sun.mail.smtp.SMTPTransport`	(optional) Specifies the service provider implementation class for transport. Allowed values are:
		`com.sun.mail.smtp.SMTPTransport`
		`com.sun.mail.smtp.SMTPSSLTransport`

`host`	none	The mail server host name.
`user`	none	The mail server user name.
`from`	none	The email address the mail server uses to indicate the message sender.
`debug`	`false`	(optional) Determines whether debugging for this resource is enabled.

TABLE 1–126 mail-resource Attributes *(Continued)*

Attribute	Default	Description
object-type	user	(optional) Defines the type of the resource. Allowed values are: ■ system-all - A system resource for all server instances and the domain application server. ■ system-admin - A system resource only for the domain application server. ■ system-instance - A system resource for all server instances only. ■ user - A user resource.
enabled	true	(optional) Determines whether this resource is enabled at runtime.

Properties

You can set properties for the mail-resource element and then get these properties in a JavaMail Session object later. Every property name must start with a mail- prefix. The Communications Application Server changes the dash (-) character to a period (.) in the name of the property, then saves the property to the MailConfiguration and JavaMail Session objects. If the name of the property doesn't start with mail-, the property is ignored.

For example, to define the property mail.password in a JavaMail Session object, first edit domain.xml as follows:

```
...
 <mail-resource jndi-name="mail/Session" ...>
     <property name="mail-password" value="adminadmin"/>
 </mail-resource>
 ...
```

After getting the JavaMail Session object, get the mail.password property to retrieve the value adminadmin, as follows:

```
String password = session.getProperty("mail.password");
```

For more information about JavaMail properties, see JavaMail API Documentation (http://java.sun.com/products/javamail/javadocs/index.html).

management-rule

Configures a self-management rule, which associates a custom self-tuning, self-configuring, or self-healing action with an event in the Communications Application Server. The action is implemented by an MBean.

Superelements

"management-rules" on page 111

Subelements

The following table describes subelements for the `management-rule` element.

TABLE 1–127 `management-rule` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element. This description is included in server log messages about the management rule. For more information on logging, see "log-service" on page 107 and "module-log-levels" on page 116.
"event" on page 63	only one	Defines the event that triggers the action associated with a management rule.
"action" on page 23	zero or one	Specifies the action of this management rule. If no action is specified, occurrence of the associated event is logged.

Attributes

The following table describes attributes for the `management-rule` element.

TABLE 1–128 `management-rule` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of this management rule.
`enabled`	`true`	(optional) If `false`, disables this management rule.

management-rules

Configures self-management rules, which associate custom self-tuning, self-configuring, and self-healing actions with events in the Communications Application Server.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `management-rules` element.

TABLE 1–129 `management-rules` Subelements

Element	Required	Description
"management-rule" on page 110	zero or more	Specifies a management rule.

Attributes

The following table describes attributes for the `management-rules` element.

TABLE 1–130 `management-rules` Attributes

Attribute	Default	Description
`enabled`	`true`	(optional) If `false`, disables all management rules. If `true`, the `enabled` attribute of each rule determines whether it is enabled.

manager-properties

Specifies session manager properties.

Superelements

"session-manager" on page 140

Subelements

The following table describes subelements for the `manager-properties` element.

TABLE 1–131 `manager-properties` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `manager-properties` element.

TABLE 1–132 `manager-properties` Attributes

Attribute	Default	Description
`session-file-name`	none; state is not preserved across restarts	(optional) Specifies the absolute or relative path to the directory in which the session state is preserved between application restarts, if preserving the state is possible. A relative path is relative to the temporary directory for this web application or SIP extension module. Applicable to the web container only if the `persistence-type` attribute of the "web-container-availability" on page 168 element is `memory`. Applicable to the SIP container if "sip-container-availability" on page 144 is disabled.
`reap-interval-in-seconds`	`60`	(optional) Specifies the time between checks for expired sessions. If the `persistence-frequency` attribute of the `web-container-availability` or "sip-container-availability" on page 144 element is set to `time-based`, active sessions are stored at this interval. Set this value lower than the frequency at which session data changes. For example, this value should be as low as possible (`1` second) for a hit counter servlet on a frequently accessed web site, or you could lose the last few hits each time you restart the server.
`max-sessions`	`-1`	(optional) Specifies the maximum number of sessions that can be in cache, or `-1` for no limit. After this, an attempt to create a new session causes an `IllegalStateException` to be thrown.
`session-id-generator-classname`	internal class generator	(optional) Not implemented. Use the `uuid-impl-class` property of the "web-container-availability" on page 168 or "sip-container-availability" on page 144 element instead.

mbean

Specifies an MBean, which implements the `javax.management.NotificationListener` interface.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `mbean` element.

TABLE 1–133 `mbean` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property. Property subelements of the `mbean` element store the names and values of attributes defined in the MBean implementation class.

Attributes

The following table describes attributes for the `mbean` element.

TABLE 1–134 `mbean` Attributes

Attribute	Default	Description
`name`	value of `impl-class-name`	The name of the MBean. The name must represent a value of a property in the `property-list` of an MBean `ObjectName`. The name is a primary key for the MBean. This is read-only.
`object-type`	`user`	(optional) Defines the type of the resource. This is read-only. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`impl-class-name`	none	Defines the fully qualified class name of the MBean implementation. This is read-only.
`object-name`	none	Defines a system-generated object name for this MBean. This is read-only.
`enabled`	`true`	(optional) Determines whether the MBean is enabled. If `false`, the MBean is not registered in the runtime environment even if the reference is enabled.

mdb-container

Configures the message-driven bean (MDB) container.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `mdb-container` element.

TABLE 1–135 `mdb-container` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `mdb-container` element.

TABLE 1–136 `mdb-container` Attributes

Attribute	Default	Description
`steady-pool-size`	`10`	(optional) Specifies the initial and minimum number of beans maintained in the pool.
`pool-resize-quantity`	`2`	(optional) Specifies the number of beans to be removed when the `idle-timeout-in-seconds` timer expires. A cleaner thread removes any unused instances. Must be `0` or greater and less than `max-pool-size` . The pool is not resized below the `steady-pool-size`.
`max-pool-size`	`60`	(optional) Specifies the maximum number of beans that can be created to satisfy client requests.
`idle-timeout-in-seconds`	`600`	(optional) Specifies the maximum time that a bean can remain idle in the pool. After this amount of time, the bean is destroyed. A value of `0` means a bean can remain idle indefinitely.

Properties

The following table describes properties for the `mdb-container` element.

TABLE 1–137 `mdb-container` Properties

Property	Default	Description
`cmt-max-runtime-exceptions`	`1`	Specifies the maximum number of `RuntimeException` occurrences allowed from a message-driven bean's `onMessage()` method when container-managed transactions are used. Deprecated.

message-security-config

Specifies configurations for message security providers.

Superelements

"security-service" on page 136

Subelements

The following table describes subelements for the `message-security-config` element.

TABLE 1–138 `message-security-config` Subelements

Element	Required	Description
"provider-config" on page 127	one or more	Specifies a configuration for one message security provider.

Attributes

The following table describes attributes for the `message-security-config` element.

TABLE 1–139 `message-security-config` Attributes

Attribute	Default	Description
`auth-layer`	`SOAP`	Specifies the message layer at which authentication is performed. The value must be `SOAP` or `HttpServlet`.
`default-provider`	none	(optional) Specifies the server provider that is invoked for any application not bound to a specific server provider.
`default-client-provider`	none	(optional) Specifies the client provider that is invoked for any application not bound to a specific client provider.

module-log-levels

Controls the level of messages logged by server subsystems to the server log. Allowed values (levels) of each subsystem attribute are, from highest to lowest: `FINEST` , `FINER`, `FINE`, `CONFIG`, `INFO`, `WARNING`, `SEVERE`, and `OFF`. Each value logs all messages for all lower values. The default value is `INFO`, which logs all `INFO`, `SEVERE` , and `WARNING` messages.

Superelements

"log-service" on page 107

Subelements

The following table describes subelements for the `module-log-levels` element.

TABLE 1–140 `module-log-levels` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `module-log-levels` element. The attribute names are the names of the Communications Application Server system loggers.

TABLE 1–141 `module-log-levels` Attributes

Attribute	Default	Description
`root`	`INFO`	(optional) Specifies the default level of messages logged by the entire Communications Application Server installation.
`server`	`INFO`	(optional) Specifies the default level of messages logged by the server instance.

TABLE 1–141 `module-log-levels` Attributes *(Continued)*

Attribute	Default	Description
`ejb-container`	`INFO`	(optional) Specifies the level of messages logged by the EJB container.
`cmp-container`	`INFO`	(optional) Specifies the level of messages logged by the CMP subsystem of the EJB container.
`mdb-container`	`INFO`	(optional) Specifies the level of messages logged by the MDB container.
`web-container`	`INFO`	(optional) Specifies the level of messages logged by the web container.
`classloader`	`INFO`	(optional) Specifies the level of messages logged by the classloader hierarchy.
`configuration`	`INFO`	(optional) Specifies the level of messages logged by the configuration subsystem.
`naming`	`INFO`	(optional) Specifies the level of messages logged by the naming subsystem.
`security`	`INFO`	(optional) Specifies the level of messages logged by the security subsystem.
`jts`	`INFO`	(optional) Specifies the level of messages logged by the Java Transaction Service.
`jta`	`INFO`	(optional) Specifies the level of messages logged by the Java Transaction API.
`admin`	`INFO`	(optional) Specifies the level of messages logged by the Administration Console subsystem.
`deployment`	`INFO`	(optional) Specifies the level of messages logged by the deployment subsystem.
`verifier`	`INFO`	(optional) Specifies the level of messages logged by the deployment descriptor verifier.
`jaxr`	`INFO`	(optional) Specifies the level of messages logged by the XML registry.
`jaxrpc`	`INFO`	(optional) Specifies the level of messages logged by the XML RPC module.
`saaj`	`INFO`	(optional) Specifies the level of messages logged by the SOAP with Attachments API for Java module.
`corba`	`INFO`	(optional) Specifies the level of messages logged by the ORB.
`javamail`	`INFO`	(optional) Specifies the level of messages logged by the JavaMail subsystem.
`jms`	`INFO`	(optional) Specifies the level of messages logged by the Java Message Service.
`connector`	`INFO`	(optional) Specifies the level of messages logged by the connector subsystem.
`jdo`	`INFO`	(optional) Specifies the level of messages logged by the Java Data Objects module.
`cmp`	`INFO`	(optional) Specifies the level of messages logged by the CMP subsystem.
`util`	`INFO`	(optional) Specifies the level of messages logged by the utility subsystem.
`resource-adapter`	`INFO`	(optional) Specifies the level of messages logged by the resource adapter subsystem.
`synchronization`	`INFO`	(optional) Specifies the level of messages logged by the synchronization subsystem.
`node-agent`	`INFO`	(optional) Specifies the level of messages logged by the node agent subsystem.
`self-management`	`INFO`	(optional) Specifies the level of messages logged by the self-management (management rules) subsystem.

TABLE 1–141 `module-log-levels` Attributes *(Continued)*

Attribute	Default	Description
`group-management-service`	`INFO`	(optional) Specifies the level of messages logged by the Group Management Service.
`management-event`	`INFO`	(optional) Specifies the level of messages logged by the self-management event subsystem.

Properties

The following table describes properties for the `module-log-levels` element. The property names are the names of the Communications Application Server system loggers. When you set these properties using the `asadmin set` command, be sure to use the escape character (double backslash `\\`) in front of the periods. For example:

```
asadmin set
config1.log-service.module-log-levels.property.javax\\.enterprise\\.system\\.container\\.clb=FINE
```

TABLE 1–142 `module-log-levels` Properties

Property	Default	Description
`javax.enterprise.system.container.sip`	`INFO`	Specifies the level of messages logged by the SIP container.
`javax.enterprise.system.container.clb`	`INFO`	Specifies the level of messages logged by the converged load balancer subsystem.
`javax.enterprise.system.container.ar`	`INFO`	Specifies the level of messages logged by the application router subsystem.
`javax.enterprise.system.container.ssr`	`INFO`	Specifies the level of messages logged by the session state replication subsystem.

module-monitoring-levels

Controls the level of monitoring of server subsystems. Allowed values of each subsystem attribute are `LOW`, `HIGH`, and `OFF`.

Superelements

"monitoring-service" on page 119

Subelements

The following table describes subelements for the `module-monitoring-levels` element.

TABLE 1–143 `module-monitoring-levels` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–144 `module-monitoring-levels` Attributes

Attribute	Default	Description
`thread-pool`	`OFF`	(optional) Specifies the level of monitoring of the thread pool subsystem.
`orb`	`OFF`	(optional) Specifies the level of monitoring of the ORB.
`ejb-container`	`OFF`	(optional) Specifies the level of monitoring of the EJB container.
`web-container`	`OFF`	(optional) Specifies the level of monitoring of the web container.
`transaction-service`	`OFF`	(optional) Specifies the level of monitoring of the transaction service.
`http-service`	`OFF`	(optional) Specifies the level of monitoring of the HTTP service.
`jdbc-connection-pool`	`OFF`	(optional) Specifies the level of monitoring of the JDBC connection pool subsystem.
`connector-connection-pool`	`OFF`	(optional) Specifies the level of monitoring of the connector connection pool subsystem.
`connector-service`	`OFF`	(optional) Specifies the level of monitoring of the connector service.
`jms-service`	`OFF`	(optional) Specifies the level of monitoring of the JMS service.
`jvm`	`OFF`	(optional) Specifies the level of monitoring of the JVM.

Properties

TABLE 1–145 `module-monitoring-levels` Properties

Attribute	Default	Description
`sip-service`	`OFF`	(optional) Specifies the level of monitoring of the SIP service.

monitoring-service

Configures the monitoring service.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `monitoring-service` element.

TABLE 1–146 `monitoring-service` Subelements

Element	Required	Description
"module-monitoring-levels" on page 118	zero or one	Controls the level of monitoring of server subsystems.
"property" on page 125	zero or more	Specifies a property or a variable.

N

node-agent

Defines a node agent, which manages server instances on a host machine.

Superelements

"node-agents" on page 121

Subelements

The following table describes subelements for the `node-agent` element.

TABLE 1–147 `node-agent` Subelements

Element	Required	Description
"jmx-connector" on page 98	zero or one	Configures a JSR 160/255 compliant remote JMX connector.
"auth-realm" on page 30	zero or one	Defines a realm for authentication.
"log-service" on page 107	only one	Configures the system logging service.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `node-agent` element.

TABLE 1–148 `node-agent` Attributes

Attribute	Default	Description
`name`	none	Specifies the node agent name.
`system-jmx-connector-name`	none	Specifies the name of the internal "jmx-connector" on page 98.
`start-servers-in-startup`	`true`	(optional) If `true`, starts all managed server instances when the node agent is started.

Properties

The following table describes properties for the `node-agent` element.

TABLE 1–149 `node-agent` Properties

Property	Default	Description
`INSTANCE-SYNC-JVM-OPTIONS`	default Communications Application Server JVM options	Sets options for the server instance synchronization JVM. Setting options that limit memory usage helps prevent `OutOfMemory` errors when large applications are synchronized or when memory is constrained. For more information, see the *Sun Java System Application Server 9.1 Administration Guide*.

node-agents

Contains node agents.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `node-agents` element.

TABLE 1–150 `node-agents` Subelements

Element	Required	Description
"node-agent" on page 120	zero or more	Defines a node agent, which manages server instances on a host machine.

O

orb

Configures the ORB.

To enable SSL for outbound connections, include an "ssl-client-config" on page 152 subelement in the parent `iiop-service` element.

Superelements

"iiop-service" on page 84

Subelements

The following table describes subelements for the `orb` element.

TABLE 1–151 `orb` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `orb` element.

TABLE 1–152 `orb` Attributes

Attribute	Default	Description
`use-thread-pool-ids`	none	Specifies a comma-separated list of `thread-pool-id` values defined in "thread-pool" on page 156 elements used by the ORB.
`message-fragment-size`	`1024`	(optional) GIOPv1.2 messages larger than this number of bytes are fragmented.
`max-connections`	`1024`	(optional) The maximum number of incoming connections on all IIOP listeners. Legal values are integers.

P

persistence-manager-factory-resource

Defines a persistence manager factory resource for container-managed persistence (CMP). Deprecated, and included for backward compatibility only. Use a "jdbc-resource" on page 94 element instead.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `persistence-manager-factory-resource` element.

TABLE 1–153 `persistence-manager-factory-resource` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `persistence-manager-factory-resource` element.

TABLE 1–154 `persistence-manager-factory-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`factory-class`	`com.sun.jdo.spi.persistence.support.sqlstore.impl.PersistenceManagerFactoryImpl`	(optional) Deprecated. Do not specify this attribute for the built-in CMP implementation.
`jdbc-resource-jndi-name`	none	Specifies the "jdbc-resource" on page 94 from which database connections are obtained. Must be the `jndi-name` of an existing `jdbc-resource`.

TABLE 1–154 `persistence-manager-factory-resource` Attributes *(Continued)*

Attribute	Default	Description
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether this resource is enabled at runtime.

principal

Contains the principal of the servlet, trusted entity, or EJB client.

Superelements

"security-map" on page 135, "trusted-entity" on page 160

Subelements

none - contains data

profiler

Configures a profiler for use with the Communications Application Server. For more information about profilers, see the *Sun Java System Application Server 9.1 Developer's Guide*.

Superelements

"java-config" on page 88

Subelements

The following table describes subelements for the `profiler` element.

TABLE 1–155 `profiler` Subelements

Element	Required	Description
"jvm-options" on page 100	zero or more	Contains profiler-specific JVM command line options.
"property" on page 125	zero or more	Specifies a property or a variable.

Note – Subelements of a `profiler` element can occur in any order.

Attributes

The following table describes attributes for the `profiler` element.

TABLE 1–156 `profiler` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the profiler.
`classpath`	none	(optional) Specifies the classpath for the profiler.
`native-library-path`	none	(optional) Specifies the native library path for the profiler.
enabled	`true`	(optional) Determines whether the profiler is enabled.

property

Specifies a property. A property adds configuration information to its parent element that is one or both of the following:

- Optional with respect to the Communications Application Server
- Needed by a system or object that the Communications Application Server doesn't have knowledge of, such as an LDAP server or a Java class

  For example, an `auth-realm` element can include `property` subelements:

```
<auth-realm name="file"
     classname="com.sun.enterprise.security.auth.realm.file.FileRealm">
     <property name="file" value="domain-dir/config/keyfile"/>
     <property name="jaas-context" value="fileRealm"/>
 </auth-realm>
```

  Which properties an `auth-realm` element uses depends on the value of the `auth-realm` element's `name` attribute. The `file` realm uses `file` and `jaas-context` properties. Other realms use different properties.

Superelements

"admin-object-resource" on page 24, "admin-service" on page 25, "alert-service" on page 26, "appclient-module" on page 27, "audit-module" on page 30, "auth-realm" on page 30, "availability-service" on page 33, "cluster" on page 36, "config" on page 38, "connector-connection-pool" on page 41, "connector-module" on page 45, "connector-resource" on page 46, "converged-lb-config" on page 48, "converged-lb-policy" on page 50, "custom-resource" on page 52, "das-config" on page 53, "diagnostic-service" on page 55, "domain" on page 56, "ejb-container" on page 58, "ejb-container-availability" on page 60, "ejb-module" on page 61, "ejb-timer-service" on page 62, "event" on page 63, "extension-module" on page 67, "external-jndi-resource" on page 69, "filter-config" on page 70, "group-management-service" on page 71, "http-listener" on page 74, "http-service" on page 79, "iiop-listener" on page 84, "j2ee-application" on page 86, "jacc-provider" on page 87, "java-config" on page 88, "jdbc-connection-pool" on page 90, "jdbc-resource" on page 94, "jms-availability" on page 94, "jms-host" on page 95, "jms-service" on page 96, "jmx-connector" on page 98, "lb-config" on page 101, "lifecycle-module" on page 103, "listener-config" on page 104, "load-balancer" on page 105, "log-service" on page 107, "mail-resource" on page 108, "management-rule" on page 110, "manager-properties" on page 112, "mbean" on page 113, "mdb-container" on page 114, "module-log-levels" on page 116, "module-monitoring-levels" on page 118, "monitoring-service" on page 119, "node-agent" on page 120, "orb" on page 122, "persistence-manager-factory-resource" on page 123, "profiler" on page 124, "provider-config" on page 127, "proxy" on page 128, "resource-adapter-config" on page 131, "security-service" on page 136, "server" on page 137, "session-properties" on page 141, "sip-container" on page 142, "sip-container-availability" on page 144, "sip-listener" on page 147, "sip-protocol" on page 148, "sip-service" on page 149, "stack-config" on page 152, "stack-layer" on page 153, "store-properties" on page 154, "transaction-service" on page 157, "trust-handler" on page 161, "virtual-server" on page 162, "web-container" on page 167, "web-container-availability" on page 168, "web-module" on page 171

Subelements

The following table describes subelements for the `property` element.

TABLE 1–157 `property` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.

Attributes

The following table describes attributes for the `property` element.

TABLE 1–158 `property` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the property or variable.
`value`	none	Specifies the value of the property or variable.

provider-config

Specifies a configuration for one message security provider.

Although the `request-policy` and `response-policy` subelements are optional, the `provider-config` element does nothing if they are not specified.

Use property subelements to configure provider-specific properties. Property values are passed to the provider when its `initialize` method is called.

Superelements

"message-security-config" on page 115

Subelements

The following table describes subelements for the `provider-config` element.

TABLE 1–159 `provider-config` Subelements

Element	Required	Description
"request-policy" on page 130	zero or one	Defines the authentication policy requirements of the authentication provider's request processing.
"response-policy" on page 134	zero or one	Defines the authentication policy requirements of the authentication provider's response processing.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `provider-config` element.

TABLE 1–160 `provider-config` Attributes

Attribute	Default	Description
`provider-id`	none	Specifies a unique identifier for this `provider-config` element.

TABLE 1–160 `provider-config` Attributes *(Continued)*

Attribute	Default	Description
`provider-type`	none	Specifies whether the provider is a `client`, `server`, or `client-server` authentication provider.
`class-name`	none	Specifies the Java implementation class of the provider. Client authentication providers must implement the `com.sun.enterprise.security.jauth.ClientAuthModule` interface. Server authentication providers must implement the `com.sun.enterprise.security.jauth.ServerAuthModule` interface. Client-server providers must implement both interfaces.

Properties

The following table describes properties for the `provider-config` element.

TABLE 1–161 `provider-config` Properties

Property	Default	Description
`security.config`	*domain-dir*`/config/wss-server-config-1.0.xml`	Specifies the location of the message security configuration file. To point to a configuration file in the *domain-dir*`/config` directory, use the prefix `${com.sun.aas.instanceRoot}/config/`, for example: `${com.sun.aas.instanceRoot}/config/wss-server-config-1.0.xml`
`debug`	`false`	If `true`, enables dumping of server provider debug messages to the server log.
`dynamic.username.password`	`false`	If `true`, signals the provider runtime to collect the user name and password from the `CallbackHandler` for each request. If `false`, the user name and password for `wsse:UsernameToken(s)` is collected once, during module initialization. This property is only applicable for a `ClientAuthModule`.
`encryption.key.alias`	`s1as`	Specifies the encryption key used by the provider. The key is identified by its `keystore` alias.
`signature.key.alias`	`s1as`	Specifies the signature key used by the provider. The key is identified by its `keystore` alias.

proxy

Specifies the proxy for converged load balancing. This provides for pass-through of HTTP and SIP requests to remote HTTP and SIP listeners, which are configured using the "http-service" on page 79 or "sip-service" on page 149 of the remote instance.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"converged-load-balancer" on page 51

Subelements

The following table describes subelements for the `proxy` element.

TABLE 1–162 `proxy` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `proxy` element.

TABLE 1–163 `proxy` Attributes

Attribute	Default	Description
`request-pool-size`	`50`	Specifies the number of request objects created and pooled by the proxy.
`send-retry-count`	`3`	Specifies the number of retries the proxy attempts with the remote instance when sending of data fails.
`read-time-out`	`1500`	Specifies in milliseconds how long the proxy waits for data from the client in the socket channel.

Properties

The following table describes properties for the `proxy` element.

TABLE 1–164 `proxy` Properties

Attribute	Default	Description
`max-parallel-connections`	none	Specifies the maximum number of outbound connections to a back-end instance.
`high-water-mark`	none	Specifies the maximum number of active outbound connections the controller handles.
`connections-to-reclaim`	none	Specifies the number of LRU connections reclaimed if the `high-water-mark` limit is reached.
`proxy-server-read-timeout`	none	Specifies the timeout for the proxy to read from the server channel once a response is available.
`socket-receive-buffer-size`	none	Specifies the proxy back-end socket receive buffer size.
`socket-send-buffer-size`	none	Specifies the proxy back-end socket send buffer size.

TABLE 1–164 `proxy` Properties *(Continued)*

Attribute	Default	Description
`client-socket-read-timeout`	none	Specifies the client socket send read timeout.

R

registry-location

Specifies the registry where web service endpoint artifacts are published.

Superelements

"web-service-endpoint" on page 172

Subelements

none

Attributes

The following table describes attributes for the `registry-location` element.

TABLE 1–165 `registry-location` Attributes

Attribute	Default	Description
`connector-resource-jndi-name`	none	Specifies the `jndi-name` of the "connector-resource" on page 46 used as the registry.

request-policy

Defines the authentication policy requirements of the authentication provider's request processing.

Superelements

"provider-config" on page 127

Subelements

none

Attributes

The following table describes attributes for the `request-policy` element.

TABLE 1–166 `request-policy` Attributes

Attribute	Default	Description
`auth-source`	none	Specifies the type of required authentication, either `sender` (user name and password) or `content` (digital signature).
`auth-recipient`	none	Specifies whether recipient authentication occurs before or after content authentication. Allowed values are `before-content` and `after-content`.

request-processing

Configures request processing threads.

Superelements

"http-service" on page 79, "sip-service" on page 149

Subelements

none

Attributes

The following table describes attributes for the `request-processing` element.

TABLE 1–167 `request-processing` Attributes

Attribute	Default	Description
`thread-count`	`5`	(optional) Specifies the maximum number of request processing threads.
`initial-thread-count`	`2`	(optional) Specifies the number of request processing threads that are available when the server starts up.
`thread-increment`	`1`	(optional) Specifies the number of request processing threads added when the number of requests exceeds the `initial-thread-count`.
`request-timeout-in-seconds`	`60`	(optional) Specifies the time at which the request times out.
`header-buffer-length-in-bytes`	`8192`	(optional) Specifies the size of the buffer used by the request processing threads to read the request data.

resource-adapter-config

Defines a connector (resource adapter) configuration. Stores configuration information for the resource adapter JavaBean in `property` subelements.

Superelements

"resources" on page 133

Subelements

The following table describes subelements for the `resource-adapter-config` element.

TABLE 1–168 `resource-adapter-config` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `resource-adapter-config` element.

TABLE 1–169 `resource-adapter-config` Attributes

Attribute	Default	Description
`name`	none	(optional) Not used. See `resource-adapter-name`.
`thread-pool-ids`	none	(optional) Specifies the `id` of a "thread-pool" on page 156 element.
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`resource-adapter-name`	none	Specifies the `name` attribute of a deployed "connector-module" on page 45. If the resource adapter is embedded in an application, then it is *app_name#rar_name*.

Properties

Properties of the `resource-adapter-config` element are the names of setter methods of the `resourceadapter-class` element in the `ra.xml` file, which defines the class name of the resource adapter JavaBean. Any properties defined here override the default values present in `ra.xml`.

resource-ref

References a resource deployed to the server instance or cluster.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"cluster" on page 36, "server" on page 137

Subelements

none

Attributes

The following table describes attributes for the `resource-ref` element.

TABLE 1–170 `resource-ref` Attributes

Attribute	Default	Description
`enabled`	`true`	(optional) Determines whether the resource is enabled.
`ref`	none	References the `name` attribute of a "custom-resource" on page 52, "external-jndi-resource" on page 69, "jdbc-resource" on page 94, "mail-resource" on page 108, "persistence-manager-factory-resource" on page 123, "admin-object-resource" on page 24 "resource-adapter-config" on page 131, "jdbc-connection-pool" on page 90, or "connector-connection-pool" on page 41 element.

resources

Contains configured resources, such as database connections, JavaMail™ sessions, and so on.

Note – You must specify a Java Naming and Directory Interface™ (JNDI) name for each resource. To avoid collisions with names of other enterprise resources in JNDI, and to avoid portability problems, all names in an Communications Application Server application should begin with the string `java:comp/env`.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `resources` element.

TABLE 1–171 `resources` Subelements

Element	Required	Description
"custom-resource" on page 52	zero or more	Defines a custom resource.
"external-jndi-resource" on page 69	zero or more	Defines a resource that resides in an external JNDI repository.
"jdbc-resource" on page 94	zero or more	Defines a JDBC (Java Database Connectivity) resource.
"mail-resource" on page 108	zero or more	Defines a JavaMail resource.
"persistence-manager-factory-resource" on page 123	zero or more	Defines a persistence manager factory resource for CMP. Deprecated. Use a "jdbc-resource" on page 94 element instead.
"admin-object-resource" on page 24	zero or more	Defines an administered object for an inbound resource adapter.
"connector-resource" on page 46	zero or more	Defines a connector (resource adapter) resource.
"resource-adapter-config" on page 131	zero or more	Defines a resource adapter configuration.
"jdbc-connection-pool" on page 90	zero or more	Defines the properties that are required for creating a JDBC connection pool.
"connector-connection-pool" on page 41	zero or more	Defines the properties that are required for creating a connector connection pool.

Note – Subelements of a `resources` element can occur in any order.

response-policy

Defines the authentication policy requirements of the authentication provider's response processing.

Superelements

"provider-config" on page 127

Subelements

none

Attributes

The following table describes attributes for the `response-policy` element.

TABLE 1–172 `response-policy` Attributes

Attribute	Default	Description
`auth-source`	none	Specifies the type of required authentication, either `sender` (user name and password) or `content` (digital signature).
`auth-recipient`	none	Specifies whether recipient authentication occurs before or after content authentication. Allowed values are `before-content` and `after-content`.

S

security-map

Maps the principal received during servlet or EJB authentication to the credentials accepted by the EIS.

Superelements

"connector-connection-pool" on page 41

Subelements

The following table describes subelements for the `security-map` element.

TABLE 1–173 `security-map` Subelements

Element	Required	Description
"principal" on page 124	one or more	Contains the principal of the servlet or EJB client.
"user-group" on page 162	one or more	Contains the group to which the principal belongs.
"backend-principal" on page 35	only one	Specifies the user name and password required by the EIS.

Attributes

The following table describes attributes for the `security-map` element.

TABLE 1–174 `security-map` Attributes

Attribute	Default	Description
`name`	none	Specifies a name for the security mapping.

security-service

Defines parameters and configuration information needed by the Java EE security service. For SSL configuration, see "ssl" on page 150. For connector module security, see "security-map" on page 135.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `security-service` element.

TABLE 1–175 `security-service` Subelements

Element	Required	Description
"auth-realm" on page 30	one or more	Defines a realm for authentication.
"jacc-provider" on page 87	one or more	Specifies a Java Authorization Contract for Containers (JACC) provider for pluggable authorization.
"audit-module" on page 30	zero or more	Specifies an optional plug-in module that implements audit capabilities.
"message-security-config" on page 115	zero or more	Specifies configurations for message security providers.
"identity-assertion-trust" on page 83	zero or more	Specifies identity assertion trust domain configuration information.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `security-service` element.

TABLE 1–176 `security-service` Attributes

Attribute	Default	Description
`default-realm`	`file`	(optional) Specifies the active authentication realm (an `auth-realm` `name` attribute) for this server instance.
`default-principal`	none	(optional) Used as the identity of the default security context when necessary and when no principal is provided. This attribute need not be set for normal server operation.
`default-principal-password`	none	(optional) The password of the default principal. This attribute need not be set for normal server operation.

TABLE 1–176 `security-service` Attributes *(Continued)*

Attribute	Default	Description
`anonymous-role`	`attribute is deprecated`	(optional) Deprecated. Do not use.
`audit-enabled`	`false`	(optional) If `true`, additional access logging is performed to provide audit information. Audit information consists of: ■ Authentication success and failure events ■ Servlet and EJB access grants and denials
`jacc`	`default`	(optional) Specifies the name of the "jacc-provider" on page 87 element to use for setting up the JACC infrastructure. Do not change the default value unless you are adding a custom JACC provider.
`audit-modules`	`default`	(optional) Specifies a space-separated list of audit provider modules used by the audit subsystem. The default value refers to the internal log-based audit module.
`activate-default-principal-to-role-mapping`	`false`	(optional) Applies a default principal for role mapping to any application that does not have an application-specific mapping defined. Every role is mapped to an instance of a `java.security.Principal` implementation class defined by `mapped-principal-class`. This class has the same name as the role.
`mapped-principal-class`	`com.sun.enterprise.deployment.Group`	(optional) Customizes the `java.security.Principal` implementation class used when `activate-default-principal-to-role-mapping` is set to `true`.

server

Defines a server instance, which is a Java EE compliant container. One server instance is specially designated as a domain administration server (DAS). The "admin-service" on page 25 subelement of the "config" on page 38 element referenced by a server's `config-ref` attribute determines whether the server is the DAS.

Note – Server instances are not the same thing as virtual servers. Each server instance is a completely separate server that contains one or more virtual servers.

Superelements

"servers" on page 139

Subelements

The following table describes subelements for the `server` element.

TABLE 1–177 `server` Subelements

Element	Required	Description
"application-ref" on page 28	zero or more	References an application or module deployed to the server instance.
"resource-ref" on page 132	zero or more	References a resource deployed to the server instance.
"system-property" on page 154	zero or more	Specifies a system property.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `server` element.

TABLE 1–178 `server` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the server instance.
`config-ref`	default "config" on page 38 element's `name`, `server-config`	(optional) References the `name` of the "config" on page 38 used by the server instance.
`node-agent-ref`	node agent created when the server instance was created	(optional) References the `name` of the "node-agent" on page 120 used by the server instance.
`lb-weight`	`100`	(optional) Specifies a server instance's relative weight for load balancing. Each server instance in a cluster has a weight, which represents the relative processing capacity of that instance. Weighted load balancing policies use this weight for load balancing requests within the cluster. It is the responsibility of the administrator to set the relative weights correctly, keeping in mind deployed hardware capacity.

server-ref

References a server instance.

Note – Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.

Superelements

"cluster" on page 36, "lb-config" on page 101, "converged-lb-config" on page 48

Subelements

The following table describes subelements for the `server-ref` element.

TABLE 1–179 `server-ref` Subelements

Element	Required	Description
"health-checker" on page 72	zero or one	Defines a health checker for the referenced server instance.

Attributes

The following table describes attributes for the `server-ref` element.

TABLE 1–180 `server-ref` Attributes

Attribute	Default	Description
`ref`	none	References the `name` attribute of a "server" on page 137 element.
`disable-timeout-in-minutes`	`30`	(optional) Specifies the time it takes this server instance to reach a quiescent state after having been disabled.
`lb-enabled`	`false`	(optional) If `true`, all load-balancers that reference this server instance consider it available to them.
`enabled`	`true`	(optional) Determines whether the server instance is enabled.

servers

Contains server instances.

Superelements

"domain" on page 56

Subelements

The following table describes subelements for the `servers` element.

TABLE 1–181 `servers` Subelements

Element	Required	Description
"server" on page 137	only one (developer profile) zero or more (cluster and enterprise profiles)	Defines a server instance.

session-config

Specifies session configuration information for the entire web or SIP container. Individual web applications or SIP extension modules can override these settings using the corresponding elements in their `sun-web.xml` or `sun-sip.xml` files.

Superelements

"web-container" on page 167, "sip-container" on page 142

Subelements

The following table describes subelements for the `session-config` element.

TABLE 1–182 `session-config` Subelements

Element	Required	Description
"session-manager" on page 140	zero or one	Specifies session manager configuration information.
"session-properties" on page 141	zero or one	Specifies session properties.

session-manager

Specifies session manager information.

Note – The session manager interface is unstable. An unstable interface might be experimental or transitional, and hence might change incompatibly, be removed, or be replaced by a more stable interface in the next release.

Superelements

"session-config" on page 140

Subelements

The following table describes subelements for the `session-manager` element.

TABLE 1–183 `session-manager` Subelements

Element	Required	Description
"manager-properties" on page 112	zero or one	Specifies session manager properties.

TABLE 1–183 `session-manager` Subelements *(Continued)*

Element	Required	Description
"store-properties" on page 154	zero or one	Specifies session persistence (storage) properties.

session-properties

Specifies session properties.

Superelements

"session-config" on page 140

Subelements

The following table describes subelements for the `session-properties` element.

TABLE 1–184 `session-properties` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–185 `session-properties` Attributes

Attribute	Default	Description
`timeout-in-seconds`	`600`	(optional) Specifies the default maximum inactive interval (in seconds) for all sessions created in this web or SIP module. If set to `0` or less, sessions in this web or SIP module never expire. If a `session-timeout` element is specified in the `web.xml` or `sip.xml` file, the `session-timeout` value overrides any `timeout-in-seconds` value. If neither `session-timeout` nor `timeout-in-seconds` is specified, the `timeout-in-seconds` default is used. Note that the `session-timeout` element in `web.xml` or `sip.xml` is specified in minutes, not seconds.

Properties

The following table describes properties for the `session-properties` element.

TABLE 1–186 `session-properties` Properties

Property	Default	Description
`enableCookies`	`true`	Uses cookies for session tracking if set to `true`.
`enableURLRewriting`	`true`	Enables URL rewriting. This provides session tracking via URL rewriting when the browser does not accept cookies. You must also use an `encodeURL` or `encodeRedirectURL` call in the servlet or JavaServer Pages™ (JSP™) page.
`idLengthBytes`	`128`	Specifies the number of bytes in this web or SIP extension module's session ID.

sip-container

Configures the Session Initiation Protocol (SIP) container.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `sip-container` element.

TABLE 1–187 `sip-container` Subelements

Element	Required	Description
"session-config" on page 140	zero or one	Specifies session configuration information for the SIP container.
"stack-config" on page 152	zero or one	Specifies the configuration of a stack of layers, typically related to protocols such as SIP.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `sip-container` element.

TABLE 1–188 `sip-container` Attributes

Attribute	Default	Description
`external-address`	determined by SIP container based on network interfaces	Specifies the address that is externally visible to clients. The clients see the entire cluster. In some cases, when providing addresses to clients, the servers must use this address to make sure the client can call back the load-balanced cluster instead of the individual server instance that provided the callback.

TABLE 1–188 `sip-container` Attributes *(Continued)*

Attribute	Default	Description
`external-sip-port`	SIP port of default "sip-listener" on page 147 (developer profile)	Specifies the SIP port that is externally visible to clients for call backs.
`external-sips-port`	SIPS port of default "sip-listener" on page 147 (developer profile)	Specifies the secure SIP (SIPS) port that is externally visible to clients for call backs.

Properties

The following table describes properties for the `sip-container` element.

TABLE 1–189 `sip-container` Properties

Property	Default	Description
`olpInserted`	`false`	If `true`, the overload protection manager is enabled in the SIP container and operates according to `CpuOverloadRegulation`, `MemOverloadRegulation`, and other `sip-container` properties. Overload regulation protects the SIP container from too high CPU usage.
`CpuOverloadRegulation`	`false`	If `true`, CPU overload protection is enabled. Applicable only if `olpInserted` is set to `true`.
`MemOverloadRegulation`	`false`	If `true`, memory overload protection is enabled. Applicable only if `olpInserted` is set to `true`.
`SampleRate`	`2`	Specifies the sample rate in seconds for updating threshold levels. Allowed values are `1` to `120` seconds.
`NumberOfSamples`	`5`	Specifies the number of consecutive samples needed before a threshold is raised. Allowed values are `2` to `20`.
`SrThreshold`	`90`	Specifies the CPU usage threshold level for SIP subsequent requests. Allowed values are `0` to `100`. Must be set higher than the `IrThreshold` value. Any `503` error responses are sent above the threshold level. Applicable only if `CpuOverloadRegulation` is set to `true`.
`IrThreshold`	`70`	Specifies the CPU usage threshold level for SIP initial requests. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `CpuOverloadRegulation` is set to `true`.
`MemIrThreshold`	`85`	Specifies the memory usage threshold level for SIP initial requests. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `MemOverloadRegulation` is set to `true`.
`HttpThreshold`	`70`	Specifies the CPU usage threshold level for HTTP requests. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `CpuOverloadRegulation` is set to `true`.

TABLE 1–189 `sip-container` Properties *(Continued)*

Property	Default	Description
`MemHttpThreshold`	`85`	Specifies the memory usage threshold level for HTTP requests. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `MemOverloadRegulation` is set to `true`.
`MmThreshold`	`90`	Specifies the CPU usage threshold level when all messages are dropped. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `CpuOverloadRegulation` is set to `true`.
`MemMmThreshold`	`99`	Specifies the memory usage threshold level when all messages are dropped. Allowed values are `0` to `100`. Any `503` error responses are sent above the threshold level. Applicable only if `MemOverloadRegulation` is set to `true`.

sip-container-availability

Enables availability in the SIP container, including replicated session persistence.

If availability is disabled, there is no high availability for session persistence. The "manager-properties" on page 112 element's `session-file-name` attribute specifies the file system location where the HTTP session state is stored if the server instance is gracefully shut down. This is useful for internal testing but is not supported for production environments.

If availability is enabled but no other `sip-container-availability` attributes are specified, the default session persistence configuration is as follows:

```
persistence-type=replicated
persistence-frequency=sip-transaction
persistence-scope=session
```

The default configuration for all applications can be changed by setting the various `sip-container-availability` attributes and properties.

You can override the various `sip-container-availability` attributes and properties for a specific application in `sun-web.xml` or `sun-sip.xml`. For details, see the *Sun Java System Application Server 9.1 Developer's Guide*.

For additional replicated session persistence properties you can set, see "availability-service" on page 33.

Superelements

"availability-service" on page 33

Subelements

The following table describes subelements for the `sip-container-availability` element.

TABLE 1–190 `sip-container-availability` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `sip-container-availability` element.

TABLE 1–191 `sip-container-availability` Attributes

Attribute	Default	Description
`availability-enabled`	`true`	(optional) If set to `true`, and if availability is enabled for the server instance (see "availability-service" on page 33), high-availability features apply to all SIP extension modules deployed to the server instance that do not have availability disabled. All instances in a cluster should have the same availability value to ensure consistent behavior.
`persistence-type`	`memory` (availability disabled) `replicated` (availability enabled)	(optional) Specifies the session persistence mechanism for SIP extension modules that have availability enabled. Allowed values are `memory` (no persistence) and `replicated` (other servers). If set to `memory`, the "manager-properties" on page 112 element's `session-file-name` attribute specifies the file system location where the session state is stored if the server instance is gracefully shut down. This is useful for internal testing but is not supported for production environments.
`persistence-frequency`	`sip-transaction`	(optional) Specifies how often the session state is stored. The default, `sip-transaction`, is the only allowed value. The session state is stored at the end of each request prior to sending a response back to the client. This provides the best guarantee that the session state is fully updated in case of failure.
`persistence-scope`	`session`	(optional) Specifies how much of the session state is stored. Allowed values are as follows: ■ `session` - The entire session state is stored every time. This mode provides the best guarantee that your session data is correctly stored for any distributable SIP extension module. ■ `modified-session` - The entire session state is stored if it has been modified. A session is considered to have been modified if `SipSession.setAttribute()` or `SipSession.removeAttribute()` was called. You must guarantee that `setAttribute()` is called every time an attribute is changed. This is not a Java EE specification requirement, but it is required for this mode to work properly. ■ `modified-attribute` - Only modified session attributes are stored. For this mode to work properly, you must follow some guidelines, which are explained immediately following this table.
`repair-during-failure`	`true`	(optional) If `true`, specifies that a forward and reverse repair should be performed on an instance that has joined or rejoined the cluster.

If the `persistence-scope` attribute is set to `modified-attribute` , your SIP extension module must follow these guidelines:

- Call `setAttribute()` every time the session state is modified.
- Make sure there are no cross-references between attributes. The object graph under each distinct attribute key is serialized and stored separately. If there are any object cross references between the objects under each separate key, they are not serialized and deserialized correctly.
- Distribute the session state across multiple attributes, or at least between a read-only attribute and a modifiable attribute.

Properties

The following table describes properties for the `sip-container-availability` element.

TABLE 1–192 `sip-container-availability` Properties

Property	Default	Description
`uuid-impl-class`	none	Specifies the name of the class that generates session IDs. If this property is not specified, the Communications Application Server's internal session ID generator is used. It is the developer's responsibility to ensure that generated IDs are universally unique even when running on multiple JVMs on multiple machines in a cluster. Failure to ensure this in the algorithm results in nondeterministic behavior and likely corruption of session data.

sip-link

Configures the SIP connection.

Superelements

"sip-protocol" on page 148

Subelements

none

Attributes

The following table describes attributes for the `sip-link` element.

TABLE 1–193 `sip-link` Attributes

Attribute	Default	Description
`connection-alive-timeout-in-seconds`	`120`	(optional) Specifies the number of seconds of inactivity allowed before the connection is closed.
`max-queue-length`	`50`	(optional) Specifies the maximum number of simultaneous write requests or connect requests, or both, that can be waiting to write on a link.
`write-timeout-in-millis`	`10`	(optional) Specifies the timeout in milliseconds for a single write operation. Allowed values are between 1 and 50.
`write-timeout-retries`	`25`	(optional) Specifies the number of retries allowed for a single write operation. Allowed values are between 1 and 25.

Properties

The following table describes properties for the `sip-link` element.

TABLE 1–194 `sip-link` Properties

Property	Default	Description
`SipLinkWaitLockTimeout`	`5000`	Specifies the maximum time a thread can wait to get an exclusive lock for a sip link.

sip-listener

Defines a SIP service listen socket. The "connection-pool" on page 40 subelement of the parent "sip-service" on page 149 element also configures some listen socket settings.

Superelements

"sip-service" on page 149

Subelements

The following table describes subelements for the `sip-listener` element.

TABLE 1–195 `sip-listener` Subelements

Element	Required	Description
"ssl" on page 150	zero or one	Defines Secure Socket Layer (SSL) parameters.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `sip-listener` element.

TABLE 1–196 `sip-listener` Attributes

Attribute	Default	Description
`id`	none	The unique listener name. A `sip-listener` name cannot begin with a number.
`address`	none	IP address of the listener. Can be in dotted-pair or IPv6 notation. Can be `any` (for `INADDR_ANY`) to listen on all IP addresses. Can be a hostname.
`port`	`5060` (non-TLS) `5061` (TLS)	Port number on which the listener listens. Legal values are `1` - `65535`. On UNIX, creating sockets that listen on ports `1` - `1024` requires superuser privileges.
`transport`	`udp_tcp`	(optional) Specifies the type of transport layer protocol. Allowed values are `udp_tcp` and `tls`.
`enabled`	`true`	(optional) Determines whether the listener is active. If set to `false`, any attempts to connect to the listener result in a socket exception (`java.net.ConnectException`).

sip-protocol

Configures SIP protocol settings.

Superelements

"sip-service" on page 149

Subelements

The following table describes subelements for the `sip-protocol` element.

TABLE 1–197 `sip-protocol` Subelements

Element	Required	Description
"sip-link" on page 146	zero or one	Configures the SIP connection.
"sip-timers" on page 150	zero or one	Configures SIP timers.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `sip-protocol` element.

TABLE 1–198 `sip-protocol` Attributes

Attribute	Default	Description
`error-response-enabled`	`false`	(optional) If `true`, responds with a `400` error code to a bad request or drop. If `false`, sends no error response.
`default-tcp-transport`	`false`	(optional) If `true`, `transport=tcp` is inserted in the URI of the `contact` and `record-route` headers.

Properties

The following table describes properties for the `sip-protocol` element.

TABLE 1–199 `sip-protocol` Properties

Property	Default	Description
`Eas503Disabled`	`false`	If `false`, the request is sent to an alternative server when a `503 Service Unavailable` error response is received, according to the RFC standard. If `true`, no alternative server is considered, so when a `503 Service Unavailable` error response is received, it is returned to the sender of the request.

sip-service

Defines the SIP service.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `sip-service` element.

TABLE 1–200 `sip-service` Subelements

Element	Required	Description
"access-log" on page 22	zero or one	Defines access log settings.
"sip-listener" on page 147	one or more	Defines a SIP listen socket.
"request-processing" on page 131	zero or one	Configures request processing threads.
"keep-alive" on page 100	zero or one	Configures keep-alive threads.
"connection-pool" on page 40	zero or one	Defines a pool of client HTTP connections.

TABLE 1–200 `sip-service` Subelements *(Continued)*

Element	Required	Description
"sip-protocol" on page 148	zero or one	Configures SIP protocol settings.
"property" on page 125	zero or more	Specifies a property or a variable.

sip-timers

Configures T1, T2, and T4 SIP timers. Detailed descriptions for the timers are in RFC 3261, Chapter 17. A summary table of SIP timers is located at `http://tools.ietf.org/html/rfc3261#page-265`.

Superelements

"sip-protocol" on page 148

Subelements

none

Attributes

The following table describes attributes for the `sip-timers` element.

TABLE 1–201 `sip-timers` Attributes

Attribute	Default	Description
`t1-in-millis`	`500`	(optional) Specifies the duration of the SIP timer T1 (round trip time estimate) in milliseconds. For unreliable transports, such as UDP, the client transaction retransmits requests at an interval that starts at T1 seconds and doubles after every retransmission. T1 is an estimate of the round-trip time (RTT). Nearly all of the SIP transaction timers scale with T1, and changing T1 adjusts their values.
`t2-in-millis`	`4000`	(optional) Specifies the duration of the SIP timer T2 (maximum retransmit interval for non-INVITE requests and INVITE responses) in milliseconds. For unreliable transports, requests are retransmitted at an interval which starts at T1 and doubles until it reaches T2. If a provisional response is received, retransmissions continue for unreliable transports, but at an interval of T2. T2 represents the amount of time a non-INVITE server transaction takes to respond to a request if it does not respond immediately.
`t4-in-millis`	`5000`	(optional) Specifies the duration of the SIP timer T4 in milliseconds. T4 represents the amount of time the network takes to clear messages between client and server transactions.

ssl

Defines SSL (Secure Socket Layer) parameters.

An `ssl` element is required inside an `http-listener` or `iiop-listener` element that has its `security-enabled` attribute set to `on` or a "sip-listener" on page 147 element with its `transport` attribute set to `tls`.

The grandparent "http-service" on page 79 element has properties that configure global SSL settings.

Superelements

"http-listener" on page 74, "sip-listener" on page 147, "iiop-listener" on page 84, "jmx-connector" on page 98, "ssl-client-config" on page 152

Subelements

none

Attributes

The following table describes attributes for the `ssl` element.

TABLE 1–202 `ssl` Attributes

Attribute	Default	Description
`cert-nickname`	`s1as`	The nickname of the server certificate in the certificate database or the PKCS#11 token. In the certificate, the name format is *tokenname*:*nickname*. Including the *tokenname*: part of the name in this attribute is optional.
`ssl2-enabled`	`false`	(optional) Determines whether SSL2 is enabled. If both SSL2 and SSL3 are enabled for a "virtual-server" on page 162, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.
`ssl2-ciphers`	none	(optional) A comma-separated list of the SSL2 ciphers used, with the prefix + to enable or - to disable, for example `+rc4` . Allowed values are `rc4`, `rc4export`, `rc2`, `rc2export`, `idea`, `des` , `desede3`.
`ssl3-enabled`	`true`	(optional) Determines whether SSL3 is enabled. The default is `true` . If both SSL2 and SSL3 are enabled for a "virtual-server" on page 162, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.
`ssl3-tls-ciphers`	none	(optional) A comma-separated list of the SSL3 ciphers used, with the prefix + to enable or - to disable, for example `+SSL_RSA_WITH_RC4_128_MD5` . Allowed values are `SSL_RSA_WITH_RC4_128_MD5`, `SSL_RSA_WITH_3DES_EDE_CBC_SHA`, `SSL_RSA_WITH_DES_CBC_SHA`, `SSL_RSA_EXPORT_WITH_RC4_40_MD5`, `SSL_RSA_WITH_NULL_MD5`, `SSL_RSA_WITH_RC4_128_SHA`, and `SSL_RSA_WITH_NULL_SHA`. Values available in previous releases are supported for backward compatibility.
`tls-enabled`	`true`	(optional) Determines whether TLS is enabled.

TABLE 1–202 `ssl` Attributes *(Continued)*

Attribute	Default	Description
`tls-rollback-enabled`	`true`	(optional) Determines whether TLS rollback is enabled. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5. For more information, see the*Sun Java System Application Server 9.1 Administration Guide*.
`client-auth-enabled`	`false`	(optional) Determines whether SSL3 client authentication is performed on every request, independent of ACL-based access control.

ssl-client-config

Defines SSL parameters for the ORB when it makes outbound SSL connections and behaves as a client.

Superelements

"iiop-service" on page 84

Subelements

The following table describes subelements for the `ssl-client-config` element.

TABLE 1–203 `ssl-client-config` Subelements

Element	Required	Description
"ssl" on page 150	only one	Defines SSL parameters.

stack-config

Specifies the configuration of a stack of layers, typically related to protocols such as SIP.

Superelements

"sip-container" on page 142

Subelements

The following table describes subelements for the `stack-config` element.

TABLE 1–204 `stack-config` Subelements

Element	Required	Description
"stack-layer" on page 153	zero or more	Specifies a layer of a stack.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `stack-config` element.

TABLE 1–205 `stack-config` Attributes

Attribute	Default	Description
`layer-order`	none	Specifies a comma-separated list of "stack-layer" on page 153 `id` values indicating the order of the stack layers.

stack-layer

Specifies a layer of a stack.

Superelements

"stack-config" on page 152

Subelements

The following table describes subelements for the `stack-layer` element.

TABLE 1–206 `stack-layer` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable. A property is a JavaBean property injected in the layer class, if a corresponding JavaBean setter exists in the layer class.

Attributes

The following table describes attributes for the `stack-layer` element.

TABLE 1–207 `stack-layer` Attributes

Attribute	Default	Description
`id`	none	Specifies a unique identifier for the `stack-layer` element.
`class-name`	none	Specifies the fully qualified name of the layer class.

store-properties

Specifies session persistence (storage) properties.

Superelements

"session-manager" on page 140

Subelements

The following table describes subelements for the `store-properties` element.

TABLE 1–208 `store-properties` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–209 `store-properties` Attributes

Attribute	Default	Description
`directory`	*domain-dir* `/generated/jsp/j2ee-apps/`*appname*`/`*appname*`_war`	(optional) Specifies the absolute or relative pathname of the directory into which individual session files are written. A relative path is relative to the temporary work directory for this web application. Applicable only if the `persistence-type` attribute of the "web-container-availability" on page 168 element is `file`.
`reap-interval-in-seconds`	`60`	(optional) Not implemented. Use the `reap-interval-in-seconds` attribute of the "manager-properties" on page 112 element instead.

system-property

Specifies a system property. A system property defines a common value for a setting at one of these levels, from highest to lowest: "domain" on page 56, "cluster" on page 36, "server" on page 137, or "config" on page 38. A value set at a higher level can be overridden at a lower level. Some system properties are predefined; see "system-property" on page 154. You can also create system properties using this element.

The following example shows the use of a predefined system property:

```
<log-service file="${com.sun.aas.instanceRoot}/logs/server.log">
    <module-log-levels admin=INFO .../>
</log-service>
```

The following example shows the creation and use of a system property:

```
<config name="config1">
    ...
    <http-service>
        ...
        <http-listener id="ls1" host="0.0.0.0" port="${ls1-port}"/>
        ...
    </http-service>
    ...
    <system-property name="ls1-port" value="8080"/>
</config>
```

Superelements

"cluster" on page 36, "config" on page 38, "domain" on page 56, "server" on page 137

Subelements

The following table describes subelements for the `system-property` element.

TABLE 1–210 `system-property` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.

Attributes

The following table describes attributes for the `system-property` element.

TABLE 1–211 `system-property` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the system property.
`value`	none	Specifies the value of the system property.

Properties

The following table lists predefined system properties.

TABLE 1–212 Predefined System Properties

Property	Default	Description
`com.sun.aas.installRoot`	depends on operating system	Specifies the directory where the Communications Application Server is installed.
`com.sun.aas.instanceRoot`	depends on operating system	Specifies the top level directory for a server instance.
`com.sun.aas.hostName`	none	Specifies the name of the host (machine).
`com.sun.aas.javaRoot`	depends on operating system	Specifies the installation directory for the Java runtime.
`com.sun.aas.imqLib`	depends on operating system	Specifies the library directory for the Sun Java System Message Queue software.
`com.sun.aas.configName`	`server-config`	Specifies the name of the "config" on page 38 used by a server instance.
`com.sun.aas.instanceName`	`server1`	Specifies the name of the server instance. This property is not used in the default configuration, but can be used to customize configuration.
`com.sun.aas.clusterName`	`cluster1`	Specifies the name of the cluster. This property is only set on clustered server instances. This property is not used in the default configuration, but can be used to customize configuration.
`com.sun.aas.domainName`	`domain1`	Specifies the name of the domain. This property is not used in the default configuration, but can be used to customize configuration.

T

thread-pool

Defines a thread pool.

Superelements

"thread-pools" on page 157

Subelements

none

Attributes

TABLE 1–213 `thread-pool` Attributes

Attribute	Default	Description
`thread-pool-id`	none	Specifies the thread pool ID.
`min-thread-pool-size`	`0`	(optional) Specifies the minimum number of threads in the pool. These are created when the thread pool is instantiated.
`max-thread-pool-size`	`200`	(optional) Specifies the maximum number of threads the pool can contain.
`idle-thread-timeout-in-seconds`	`120`	(optional) Specifies the amount of time after which idle threads are removed from the pool.
`num-work-queues`	`1`	(optional) Specifies the total number of work queues serviced by this thread pool.

thread-pools

Contains thread pools.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `thread-pools` element.

TABLE 1–214 `thread-pools` Subelements

Element	Required	Description
"thread-pool" on page 156	one or more	Defines a thread pool.

transaction-service

Configures the Java Transaction Service (JTS).

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `transaction-service` element.

TABLE 1–215 `transaction-service` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `transaction-service` element.

TABLE 1–216 `transaction-service` Attributes

Attribute	Default	Description
`automatic-recovery`	`false` (developer profile) `true` (cluster and enterprise profiles)	(optional) If `true`, the server instance attempts transaction recovery during startup.
`timeout-in-seconds`	`0`	(optional) Specifies the amount of time after which the transaction is aborted. If set to `0`, the transaction never times out.
`tx-log-dir`	directory specified by the `log-root` attribute of the "domain" on page 56 element	(optional) Specifies the parent directory of the transaction log directory *instance-name*`/tx`. The directory in which the transaction logs are kept must be writable by the user account under which the server runs. A relative path is relative to the `log-root` attribute of the "domain" on page 56 element.
`heuristic-decision`	`rollback`	(optional) If the outcome of a distributed transaction cannot be determined because other participants are unreachable, this property determines the outcome. Allowed values are `rollback` and `commit`.
`retry-timeout-in-seconds`	`600`	(optional) Determines the retry time in the following scenarios: ■ At the transaction recovery time, if resources are unreachable. ■ If there are any transient exceptions in the second phase of a two phase commit protocol. A negative value specifies infinite retries. A value of `0` (zero) specifies no retries. A positive value indicates the time after which a retry is attempted.
`keypoint-interval`	`65536` (64 K)	(optional) Specifies the number of transactions between keypoint operations in the log. Keypoint operations reduce the size of the transaction log file by compressing it. A larger value for this attribute results in a larger transaction log file, but fewer keypoint operations and potentially better performance. A smaller value results in smaller log files, but slightly reduced performance due to the greater frequency of keypoint operations.

Properties

The following table describes properties for the `transaction-service` element.

TABLE 1–217 `transaction-service` Properties

Property	Default	Description
`oracle-xa-recovery-workaround`	`true`	If `true`, the Oracle XA Resource workaround is used in transaction recovery.
`disable-distributed-transaction-logging`	`false`	If `true`, disables transaction logging, which might improve performance. If the `automatic-recovery` attribute is set to `true` , this property is ignored.
`xaresource-txn-timeout`	specific to the `XAResource` used	Changes the `XAResource` timeout. In some cases, the `XAResource` default timeout can cause transactions to be aborted, so it is desirable to change it.
`pending-txn-cleanup-interval`	none if this property is absent, `60` if this property is present but has no value	Specifies the interval, in seconds, at which an asynchronous thread checks for pending transactions and completes them.
`use-last-agent-optimization`	`true`	If `true`, enables last agent optimization, which improves the throughput of transactions. If one non-XA resource is used with XA resources in the same transaction, the non XA resource is the last agent.

`delegated-recovery`	`false`	If `true`, cluster-wide delegated recovery is enabled. You must also facilitate storing of transaction logs in a shared file system in one of these ways:
		■ Set the "domain" on page 56 element's `log-root` attribute to a shared file system base directory and set the `tx-log-dir` attribute to a relative path.
		■ Set `tx-log-dir` to an absolute path to a shared file system directory, in which case `log-root` is ignored for transaction logs.
		■ Set a "system-property" on page 154 called `TX-LOG-DIR` to a shared file system directory. For example:
		`<server config-ref="server-config" name="server">`
		`<system-property name="TX-LOG-DIR"`
		`value="/net/tulsa/nodeagents/na/instance1/logs"/>`
		`</server>`

`wait-time-before-recovery-insec`	`60`	Specifies the wait time, in seconds, after which an instance starts the recovery for a dead instance.
`db-logging-resource`	none	Specifies the JNDI name of the JDBC resource for the database to which transactions are logged. For more information, see Chapter 16, "Using the Transaction Service," in *Sun Java System Application Server 9.1 Developer's Guide*.

transformation-rule

Configures an e**X**tensible **S**tylesheet **L**anguage **T**ransformation (XSLT) rule, which transforms a web service message.

Superelements

"web-service-endpoint" on page 172

Subelements

The following table describes subelements for the `transformation-rule` element.

TABLE 1–218 `transformation-rule` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.

Attributes

The following table describes attributes for the `transformation-rule` element.

TABLE 1–219 `transformation-rule` Attributes

Attribute	Default	Description
`name`	none	The name of the rule.
`enabled`	`true`	(optional) Determines whether the rule is enabled.
`apply-to`	`request`	(optional) Specifies whether the rule is applied to the request, the response, or both. Allowed values are: ■ `request` - Transformations are applied to the request in the order specified. ■ `response` - Transformations are applied to the response in the order specified. ■ `both` - Transformations are applied to both the request and the response. The order is reversed for the response.
`rule-file-location`	*domain-dir*`/generated/xml/`*app-or-module*`/`*xslt-file*	A fully qualified or relative path to the rule file that performs the transformation. Only XSLT files are allowed.

trusted-entity

Specifies intermediate hosts and domains according to RFC 3325.

Superelements

"identity-assertion-trust" on page 83

Subelements

The following table describes subelements for the `trusted-entity` element.

TABLE 1–220 `trusted-entity` Subelements

Element	Required	Description
"ip-address" on page 85	only one	Identifies the trusted host on the network using an IP address.
"host-name" on page 57	zero or one	Identifies the trusted host on the network using domain names.
"principal" on page 124	zero or one	Identifies the principal of the trusted entity.

Attributes

The following table describes attributes for the `trusted-entity` element.

TABLE 1–221 `trusted-entity` Attributes

Attribute	Default	Description
`id`	none	Specifies a unique identifier for the `trusted-entity` element.
`trusted-as`	none	(optional) When set to `intermediate`, specifies configuration information for incoming messages. When set to `destination`, specifies configuration information for outgoing messages. When not set, specifies configuration information for both incoming and outgoing messages.

trust-handler

Specifies a custom trust handler according to RFC 3325. This provides a custom implementation to determine trust and to convert user identity to a format recognized by the system.

Superelements

"identity-assertion-trust" on page 83

Subelements

The following table describes subelements for the `trust-handler` element.

TABLE 1–222 `trust-handler` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `trust-handler` element.

TABLE 1–223 `trust-handler` Attributes

Attribute	Default	Description
`class-name`	none	Specifies the name of the class that implements `com.sun.enterprise.security.trust.TrustHandler`.

U

user-group

Contains the group to which the principal belongs.

Superelements

"security-map" on page 135

Subelements

none - contains data

V

virtual-server

Defines a virtual server. A virtual server, also called a virtual host, is a virtual web server that serves content targeted for a specific URL. Multiple virtual servers can serve content using the same or different host names, port numbers, or IP addresses. The HTTP service can direct incoming web requests to different virtual servers based on the URL.

When the Communications Application Server is first installed, a default virtual server is created. (You can also assign a default virtual server to each new "http-listener" on page 74 you create.)

Note – Virtual servers are not the same thing as server instances. Each server instance is a completely separate server that contains one or more virtual servers.

Before the Communications Application Server can process a request, it must accept the request via a listener, then direct the request to the correct virtual server. The virtual server is determined as follows:

- If the listener is configured to only a default virtual server, that virtual server is selected.
- If the listener has more than one virtual server configured to it, the request `Host` header is matched to the `hosts` attribute of a virtual server. If no `Host` header is present or no `hosts` attribute matches, the default virtual server for the listener is selected.

If a virtual server is configured to an SSL listener, its `hosts` attribute is checked against the subject pattern of the certificate at server startup, and a warning is generated and written to the server log if they don't match.

Superelements

"http-service" on page 79

Subelements

The following table describes subelements for the `virtual-server` element.

TABLE 1–224 `virtual-server` Subelements

Element	Required	Description
"http-access-log" on page 73	zero or one	Defines an access log file.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `virtual-server` element.

TABLE 1–225 `virtual-server` Attributes

Attribute	Default	Description
`id`	none	Virtual server ID. This is a unique ID that allows lookup of a specific virtual server. A virtual server ID cannot begin with a number.
`http-listeners`	none	(optional) In a comma-separated list, references `id` attributes of "http-listener" on page 74 elements that specify the connection(s) the virtual server uses. Required only for a `virtual-server` that is not referenced by the `default-virtual-server` attribute of an `http-listener`.
`default-web-module`	none	(optional) References the `name` attribute of the default "web-module" on page 171 for this virtual server, which responds to requests that cannot be resolved to other web modules deployed to this virtual server (see the "application-ref" on page 28 element).
`hosts`	none	A comma-separated list of values, each of which selects the current virtual server when included in the `Host` request header. Two or more `virtual-server` elements that reference or are referenced by the same `http-listener` cannot have any `hosts` values in common.

TABLE 1–225 `virtual-server` Attributes *(Continued)*

Attribute	Default	Description
`state`	`on`	(optional) Determines whether a `virtual-server` is active (`on`) or inactive (`off`, `disabled`). The default is `on` (active). When inactive, a `virtual-server` does not service requests. If a `virtual-server` is `disabled`, only the global server administrator can turn it `on`.
`docroot`	none	(optional) Specifies the document root for this virtual server.
`log-file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 56 element	(optional) Writes this virtual server's log messages to a log file separate from the server log. The file and directory in which the virtual server log is kept must be writable by the user account under which the server runs. See the "log-service" on page 107 description for details about logs.

Properties

The following table describes properties for the `virtual-server` element.

TABLE 1–226 `virtual-server` Properties

Property	Default	Description
`sso-enabled`	`false` (developer and cluster profiles) `true` (enterprise profile)	If `true`, single sign-on is enabled for web applications on this virtual server that are configured for the same realm. If `false`, single sign-on is disabled for this virtual server, and users must authenticate separately to every application on the virtual server. This setting overrides the `sso-enabled` property setting of the "http-service" on page 79 element.
`sso-max-inactive-seconds`	`300`	Specifies the time after which a user's single sign-on record becomes eligible for purging if no client activity is received. Since single sign-on applies across several applications on the same virtual server, access to any of the applications keeps the single sign-on record active. Higher values provide longer single sign-on persistence for the users at the expense of more memory use on the server.
`sso-reap-interval-seconds`	`60`	Specifies the interval between purges of expired single sign-on records.
`setCacheControl`	none	Specifies a comma-separated list of `Cache-Control` response directives. For a list of valid directives, see section 14.9 of the document at http://www.ietf.org/rfc/rfc2616.txt.
`accessLoggingEnabled`	`false` (developer and cluster profiles) `true` (enterprise profile)	If `true`, enables access logging for this virtual server only. If `false`, disables access logging for this virtual server only.

TABLE 1–226 `virtual-server` Properties *(Continued)*

Property	Default	Description
`accessLogBuffer Size`	`32768`	Specifies the size, in bytes, of the buffer where access log calls are stored. If the value is less than `5120`, a warning message is issued, and the value is set to `5120`. To set this property for all virtual servers, set it as a property of the parent "http-service" on page 79 element.
`accessLogWriter Interval`	`300`	Specifies the number of seconds before the log is written to the disk. The access log is written when the buffer is full or when the interval expires. If the value is `0`, the buffer is always written even if it is not full. This means that each time the server is accessed, the log message is stored directly to the file. To set this property for all virtual servers, set it as a property of the parent "http-service" on page 79 element.
`allowRemoteAddress`	none	Specifies a comma-separated list of regular expression patterns that the remote client's IP address is compared to. If this property is specified, the remote address *must* match for this request to be accepted. If this property is not specified, all requests are accepted *unless* the remote address matches a `denyRemoteAddress` pattern.
`denyRemoteAddress`	none	Specifies a comma-separated list of regular expression patterns that the remote client's IP address is compared to. If this property is specified, the remote address must *not* match for this request to be accepted. If this property is not specified, request acceptance is governed solely by the `allowRemoteAddress` property.
`allowRemoteHost`	none	Specifies a comma-separated list of regular expression patterns that the remote client's hostname (as returned by `[java.net.]Socket.getInetAddress().getHostName()`) is compared to. If this property is specified, the remote hostname *must* match for this request to be accepted. If this property is not specified, all requests are accepted *unless* the remote hostname matches a `denyRemoteHost` pattern.
`denyRemoteHost`	none	Specifies a comma-separated list of regular expression patterns that the remote client's hostname (as returned by `[java.net.]Socket.getInetAddress().getHostName()`) is compared to. If this property is specified, the remote hostname must *not* match for this request to be accepted. If this property is not specified, request acceptance is governed solely by the `allowRemoteHost` property.
`authRealm`	none	Specifies the `name` attribute of an "auth-realm" on page 30 element, which overrides the server instance's default realm for stand-alone web applications deployed to this virtual server. A realm defined in a stand-alone web application's `web.xml` file overrides the virtual server's realm.
`securePages WithPragma`	`true`	Set this property to `false` to ensure that for all web applications on this virtual server file downloads using SSL work properly in Internet Explorer. You can set this property for a specific web application. For details, see "sun-web-app" in *Sun Java System Application Server 9.1 Application Deployment Guide*.

TABLE 1–226 `virtual-server` Properties *(Continued)*

Property	Default	Description

`alternatedocroot_n`	none	Specifies an alternate document root (docroot), where *n* is a positive integer that allows specification of more than one. Alternate docroots allow web applications to serve requests for certain resources from outside their own docroot, based on whether those requests match one (or more) of the URI patterns of the web application's alternate docroots.
		
		If a request matches an alternate docroot's URI pattern, it is mapped to the alternate docroot by appending the request URI (minus the web application's context root) to the alternate docroot's physical location (directory). If a request matches multiple URI patterns, the alternate docroot is determined according to the following precedence order:
		■ Exact match
		■ Longest path match
		■ Extension match
		
		For example, the following properties specify three alternate docroots. The URI pattern of the first alternate docroot uses an exact match, whereas the URI patterns of the second and third alternate docroots use extension and longest path prefix matches, respectively.
		
		`<property name="alternatedocroot_1"`
		`value="from=/my.jpg dir=/srv/images/jpg"/>`
		`<property name="alternatedocroot_2"`
		`value="from=*.jpg dir=/srv/images/jpg"/>`
		`<property name="alternatedocroot_3"`
		`value="from=/jpg/* dir=/src/images"/>`
		
		The `value` of each alternate docroot has two components: The first component, `from`, specifies the alternate docroot's URI pattern, and the second component, `dir`, specifies the alternate docroot's physical location (directory). Spaces are allowed in the `dir` component.
		
		You can set this property for a specific web application. For details, see "sun-web-app" in *Sun Java System Application Server 9.1 Application Deployment Guide*.

`contextXmlDefault`	none	Specifies the location, relative to *domain-dir*, of the `context.xml` file for this virtual server, if one is used. For more information about the `context.xml` file, see The Context Container (`http://tomcat.apache.org/tomcat-5.5-doc/config/context.html`).
`allowLinking`	`false`	If `true`, resources that are symbolic links in web applications on this virtual server are served. The value of this property in the `sun-web.xml` file takes precedence if defined. For details, see "sun-web-app" in *Sun Java System Application Server 9.1 Application Deployment Guide*. **Caution –** Setting this property to `true` on Windows systems exposes JSP source code.

TABLE 1–226 `virtual-server` Properties *(Continued)*

Property	Default	Description

`send-error_`*n*	none	Specifies custom error page mappings for the virtual server, which are inherited by all web applications deployed on the virtual server. A web application can override these custom error page mappings in its `web.xml` deployment descriptor. The value of each `send-error_`*n* property has three components, which may be specified in any order:
		
		The first component, `code`, specifies the three-digit HTTP response status code for which the custom error page should be returned in the response.
		
		The second component, `path`, specifies the absolute or relative file system path of the custom error page. A relative file system path is interpreted as relative to the *domain-dir*`/config` directory.
		
		The third component, `reason`, is optional and specifies the text of the reason string (such as `Unauthorized` or `Forbidden`) to be returned.
		
		For example:
		
		`<property name="send-error_1"`
		`value="code=401 path=/myhost/401.html reason=MY-401-REASON"/>`
		
		This example property definition causes the contents of `/myhost/401.html` to be returned with 401 responses, along with this response line:
		
		`HTTP/1.1 401 MY-401-REASON`

`redirect_`*n*	none	Specifies that a request for an old URL is treated as a request for a new URL. These properties are inherited by all web applications deployed on the virtual server. The value of each `redirect_`*n* property has two components, which may be specified in any order:
		
		The first component, `from`, specifies the prefix of the requested URI to match.
		
		The second component, `url-prefix`, specifies the new URL prefix to return to the client. The from prefix is simply replaced by this URL prefix.
		
		For example:
		
		`<property name="redirect_1"`
		`value="from=/dummy url-prefix=http://etude"/>`

W

web-container

Configures the web container.

Superelements

"config" on page 38

Subelements

The following table describes subelements for the `web-container` element.

TABLE 1–227 `web-container` Subelements

Element	Required	Description
"session-config" on page 140	zero or one	Specifies session configuration information for the web container.
"property" on page 125	zero or more	Specifies a property or a variable.

Properties

The following table describes properties for the `web-container` element.

TABLE 1–228 `web-container` Properties

Property	Default	Description
`dispatcher-max-depth`	`20`	Prevents recursive `include` or `forward` statements from creating an infinite loop by setting a maximum nested dispatch level. If this level is exceeded, the following message is written to the server log: `Exceeded maximum depth for nested request dispatches`

web-container-availability

Enables availability in the web container, including HTTP session persistence.

If availability is disabled, there is no high availability for HTTP session persistence. In other words, `persistence-type=memory`.

If availability is enabled but no other `web-container-availability` attributes are specified, the default session persistence configuration is as follows:

```
persistence-type=replicated
persistence-frequency=time-based
persistence-scope=session
```

The default configuration for all applications can be changed by setting the various `web-container-availability` attributes and properties.

You can override the various `web-container-availability` attributes and properties for a specific application in `sun-web.xml`. For details, see the *Sun Java System Application Server 9.1 Developer's Guide*.

For additional replicated session persistence properties you can set, see "availability-service" on page 33.

Superelements

"availability-service" on page 33

Subelements

The following table describes subelements for the `web-container-availability` element.

TABLE 1–229 `web-container-availability` Subelements

Element	Required	Description
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `web-container-availability` element.

TABLE 1–230 `web-container-availability` Attributes

Attribute	Default	Description
`availability-enabled`	`true`	(optional) If set to `true`, and if availability is enabled for the server instance (see "availability-service" on page 33), high-availability features apply to all web applications deployed to the server instance that do not have availability disabled. All instances in a cluster should have the same availability value to ensure consistent behavior.
`persistence-type`	`memory` (availability disabled) `replicated` (availability enabled)	(optional) Specifies the session persistence mechanism for web applications that have availability enabled. Allowed values are `memory` (no persistence), `file` (the file system) and `replicated` (other servers). If set to `memory`, the "manager-properties" on page 112 element's `session-file-name` attribute specifies the file system location where the HTTP session state is stored if the server instance is gracefully shut down. This is useful for internal testing but is not supported for production environments. If set to `file`, the "store-properties" on page 154 element's `directory` attribute specifies the file system location where the HTTP session state is stored. Persisting to the file system is useful for internal testing but is not supported for production environments.

TABLE 1–230 `web-container-availability` Attributes *(Continued)*

Attribute	Default	Description
`persistence-frequency`	`web-method`	(optional) Specifies how often the session state is stored. Applicable only if the `persistence-type` is `replicated`. Allowed values are as follows: ■ `web-method` - The session state is stored at the end of each web request prior to sending a response back to the client. This mode provides the best guarantee that the session state is fully updated in case of failure. ■ `time-based` - The session state is stored in the background at the frequency set by the "manager-properties" on page 112 element's `reap-interval-in-seconds` attribute. This mode provides less of a guarantee that the session state is fully updated. However, it can provide a significant performance improvement because the state is not stored after each request.
`persistence-scope`	`session`	(optional) Specifies how much of the session state is stored. Applicable only if the `persistence-type` is `replicated`. Allowed values are as follows: ■ `session` - The entire session state is stored every time. This mode provides the best guarantee that your session data is correctly stored for any distributable web application. ■ `modified-session` - The entire session state is stored if it has been modified. A session is considered to have been modified if `HttpSession.setAttribute()` or `HttpSession.removeAttribute()` was called. You must guarantee that `setAttribute()` is called every time an attribute is changed. This is not a Java EE specification requirement, but it is required for this mode to work properly. ■ `modified-attribute` - Only modified session attributes are stored. For this mode to work properly, you must follow some guidelines, which are explained immediately following this table.
`sso-failover-enabled`	`false`	(optional) If `true`, the single sign-on state is highly available. To enable single sign-on, use the `sso-enabled` property of the "virtual-server" on page 162 element.

If the `persistence-scope` attribute is set to `modified-attribute` , your web application must follow these guidelines:

- Call `setAttribute()` every time the session state is modified.
- Make sure there are no cross-references between attributes. The object graph under each distinct attribute key is serialized and stored separately. If there are any object cross references between the objects under each separate key, they are not serialized and deserialized correctly.
- Distribute the session state across multiple attributes, or at least between a read-only attribute and a modifiable attribute.

Properties

The following table describes properties for the `web-container-availability` element.

TABLE 1–231 `web-container-availability` Properties

Property	Default	Description
`uuid-impl-class`	none	Specifies the name of the class that generates session IDs. If this property is not specified, the Communications Application Server's internal session ID generator is used. It is the developer's responsibility to ensure that generated IDs are universally unique even when running on multiple JVMs on multiple machines in a cluster. Failure to ensure this in the algorithm results in nondeterministic behavior and likely corruption of HTTP session data.

web-module

Specifies a deployed web module.

Superelements

"applications" on page 29

Subelements

The following table describes subelements for the `web-module` element.

TABLE 1–232 `web-module` Subelements

Element	Required	Description
"description" on page 54	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 172	zero or more	Configures a web service endpoint.
"property" on page 125	zero or more	Specifies a property or a variable.

Attributes

The following table describes attributes for the `web-module` element.

TABLE 1–233 `web-module` Attributes

Attribute	Default	Description
`name`	none	The name of the web module.

TABLE 1–233 `web-module` Attributes *(Continued)*

Attribute	Default	Description
`context-root`	none	The context root at which the web module is deployed. The context root can be the empty string or just `/`. The context root can start with the `/` character, but doesn't have to. For load balancing to work, web module context roots must be unique within a cluster. See the *Sun Java System Application Server 9.1 High Availability Administration Guide* for more information about load balancing. **Note –** Some topics in the documentation pertain to features that are available only in domains that are configured to support clusters. Examples of domains that support clusters are domains that are created with the cluster profile or the enterprise profile. For information about profiles, see "Usage Profiles" in *Sun Java System Application Server 9.1 Administration Guide*.
`location`	none	A fully qualified or relative path to the directory to which the contents of the `.war` file have been extracted. If relative, it is relative to the following directory: *domain-dir*`/applications/j2ee-modules/`
`object-type`	`user`	(optional) Defines the type of the resource. Allowed values are: ■ `system-all` - A system resource for all server instances and the domain application server. ■ `system-admin` - A system resource only for the domain application server. ■ `system-instance` - A system resource for all server instances only. ■ `user` - A user resource.
`enabled`	`true`	(optional) Determines whether the web module is enabled.
`libraries`	none	(optional) Specifies an absolute or relative path to libraries specific to this module or application. A relative path is relative to *domain-dir*`/lib/applibs`. If the path is absolute, the path must be accessible to the domain administration server (DAS), which means it must be under *domain-dir*. To include more than one path, use a system-specific separator, such as a colon for Solaris or a semicolon for Windows. The libraries are made available to the application in the order in which they are specified.
`availability-enabled`	`false`	(optional) Specifies whether availability is enabled in this web application for HTTP session persistence (and potentially passivation). Availability must also be enabled for the application or stand-alone web module during deployment. For more information about availability, see "availability-service" on page 33.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed as a directory.

web-service-endpoint

Configures a web service endpoint, which can be a JAX-RPC/JAXWS 2.0 or JSR-109 web service.

Superelements

"ejb-module" on page 61, "j2ee-application" on page 86, "web-module" on page 171

Subelements

The following table describes subelements for the `web-service-endpoint` element.

TABLE 1–234 `web-service-endpoint` Subelements

Element	Required	Description
"registry-location" on page 130	zero or more	Specifies the registry where web service endpoint artifacts are published.
"transformation-rule" on page 159	zero or more	Configures an e**X**tensible **S**tylesheet **L**anguage **T**ransformation (XSLT) rule.

Attributes

The following table describes attributes for the `web-service-endpoint` element.

TABLE 1–235 `web-service-endpoint` Attributes

Attribute	Default	Description
`name`	none	The fully qualified name of the web service. For a web service endpoint within an application, the format is as follows: *module-name#endpoint-name* For example: `jaxrpc-simple.war#HelloIF` For a web service endpoint that is a stand-alone module, the name is just the *endpoint-name*.
`monitoring`	`OFF`	(optional) Specifies the monitoring level for this web service. For information about monitoring levels, see "module-monitoring-levels" on page 118.
`max-history-size`	`25`	(optional) Specifies the maximum number of monitoring records stored for this endpoint.
`jbi-enabled`	`false`	(optional) Determines whether the visibility of this endpoint as a Java Business Integration service is enabled or disabled.

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X