Not For Publication

Sun Java System Application Server Platform Edition 9 Administration Reference

Beta

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

Part No: 819–3661

Contents

Preface

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

Who Should Use This Book

This *Administration Reference* is intended for use by administrators and software developers who maintain and use Sun Java System servers and software. Application Server administrators and software developers should already understand the following technologies:

- Java technology
- The Java Platform, Enterprise Edition (Java EE platform), version 5
- Extensible Markup Language (XML)

How This Book Is Organized

This book contains one chapter, Chapter 1, which describes in detail the structure and content of this file. The file's elements are listed in alphabetical order.

Application Server Documentation Set

The Application Server documentation set describes deployment planning and system installation. The Uniform Resource Locator (URL) for stand-alone Application Server documentation is `http://docs.sun.com/app/docs/coll/1343.3`. For an introduction to Application Server, refer to the books in the order in which they are listed in the following table.

TABLE P–1 Books in the Application Server Documentation Set

Book Title	Description
Documentation Center	Application Server documentation topics organized by task and subject.

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

Book Title	Description
Release Notes	Late-breaking information about the software and the documentation. Includes a comprehensive, table-based summary of the supported hardware, operating system, Java Development Kit (JDK™), and database drivers.
Quick Start Guide	How to get started with the Application Server product.
Installation Guide	Installing the software and its components.
Application Deployment Guide	Deployment of applications and application components to the Application Server. Includes information about deployment descriptors.
Developer's Guide	Creating and implementing Java Platform, Enterprise Edition (Java EE platform) applications intended to run on the Application Server that follow the open Java standards model for Java EE components and APIs. Includes information about developer tools, security, debugging, and creating lifecycle modules.
Java EE 5 Tutorial	Using Java EE 5 platform technologies and APIs to develop Java EE applications.
Administration Guide	Configuring, managing, and deploying Application Server subsystems and components from the Admin Console.
Administration Reference	Editing the Application Server configuration file, `domain.xml`.
Upgrade and Migration Guide	Migrating your applications to the new Application Server programming model, specifically from Application Server 6.x, and 7.x, and 8.x. This guide also describes differences between adjacent product releases and configuration options that can result in incompatibility with the product specifications.
Troubleshooting Guide	Solving Application Server problems.
Error Message Reference	Solving Application Server error messages.
Reference Manual	Utility commands available with the Application Server; written in man page style. Includes the `asadmin` command line interface.

Related Books

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

- Message Queue documentation (`http://docs.sun.com/app/docs/coll/1343.3`)
- Directory Server documentation (`http://docs.sun.com/app/docs/coll/1316.1`)
- Web Server documentation (`http://docs.sun.com/app/docs/coll/1308.1`)

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
install-dir	Represents the base installation directory for Application Server.	Solaris™ and Linux operating system installations, non-root user: *user's-home-directory*`/SUNWappserver` Solaris and Linux installations, root user: `/opt/SUNWappserver` Windows, all installations: *SystemDrive*`:\Sun\AppServer`
domain-root-dir	Represents the directory containing all domains.	*install-dir*`/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*

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*.

TABLE P–3 Typographic Conventions *(Continued)*

Typeface	Meaning	Example
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.

Accessing Sun Resources Online

The docs.sun.com^{SM} web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. Books are available as online files in PDF and HTML formats. Both formats are readable by assistive technologies for users with disabilities.

To access the following Sun resources, go to `http://www.sun.com`:

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research

- Communities (for example, Sun Developer Network)

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 819-3661.

◆ ◆ ◆ 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 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 Application Server deployment descriptors override corresponding settings in the `domain.xml` file unless otherwise stated. For more information about the Application Server deployment descriptors, see the *Sun Java System Application Server Platform Edition 9 Application Deployment Guide*.

The sun-domain_1_2.dtd File

The `sun-domain_1_2.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_2.dtd` file is located in the *install-dir*`/lib/dtds` directory.

Note – Do not edit the `sun-domain_1_2.dtd` file; its contents change only with new versions of the Application Server.

The `sun-domain_1_2.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_2.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 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.

Variables

Variables and variable references are needed for two reasons:

- Parts of the Application Server share much configuration information but differ in specific details.
- 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 104 element or the "jvm-options" on page 70 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 98, "config" on page 28, and "domain" on page 40 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 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 → 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 `config` element that references that specific `server`
- In the `jvm-options` subelements of the `config` element referenced by that `server`

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 24 element references an application or module that is deployed to its parent "server" on page 98 element. The `application-ref` element's `ref` attribute has the same value as the `name` attribute of a "lifecycle-module" on page 71, "j2ee-application" on page 59, "ejb-module" on page 43, "web-module" on page 114, "connector-module" on page 34, or "appclient-module" on page 23 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 88 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
.    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
.    .    .    http-service     P
.    .    .    .    http-listener     P
.    .    .    .    .    ssl
.    .    .    .    virtual-server     P
.    .    .    iiop-service
.    .    .    .    orb     P
.    .    .    .    ssl-client-config
.    .    .    .    .    ssl
.    .    .    .    iiop-listener     P
.    .    .    .    .    ssl
.    .    .    admin-service     P
.    .    .    .    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
.    .    .    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
.    .    .    transaction-service     P
.    .    .    monitoring-service     P
.    .    .    .    module-monitoring-levels     P
.    .    .    diagnostic-service     P
```

```
.   .   .   java-config     P
.   .   .   .   profiler     P
.   .   .   .   .   jvm-options
.   .   .   .   jvm-options
.   .   .   thread-pools
.   .   .   .   thread-pool
.   .   .   alert-service     P
.   .   .   .   alert-subscription
.   .   .   .   .   listener-config     P
.   .   .   .   .   filter-config     P
.   .   .   system-property
.   .   .   .   description
.   servers
.   .   server     P
.   .   .   application-ref
.   .   .   resource-ref
.   .   .   system-property
.   .   .   .   description
.   system-property
.   .   description
```

Alphabetical List of Elements

A

access-log

Defines access log settings for each "http-access-log" on page 51 subelement of each "virtual-server" on page 110.

Superelements

"http-service" on page 55

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.
`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`	(optional) Specifies the time interval between log rotations if `rotation-policy` is set to `time`.
`rotation-suffix`	`yyyy-MM-dd`	(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 76

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 79 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 95

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 39	zero or one	Contains a text description of this element.
"property" on page 88	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 34 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.

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

Attribute	Default	Description
`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 2006Q2 Administration Guide*.

admin-service

Determines whether the server instance is a regular instance, a domain administration server, or a combination. In the Platform Edition, there is only one server instance, and it is a combination.

Superelements

"config" on page 28

Subelements

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

TABLE 1–5 `admin-service` Subelements

Element	Required	Description
"das-config" on page 37	only one	Defines a domain administration server configuration.
"property" on page 88	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`	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. For the Platform Edition, the default is the only value allowed.

alert-service

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

Superelements

"config" on page 28

Subelements

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

TABLE 1–7 `alert-service` Subelements

Element	Required	Description
"alert-subscription" on page 22	zero or more	Configures a subscription to system status alerts.
"property" on page 88	zero or more	Specifies a property or a variable.

alert-subscription

Configures a subscription to system status alerts.

Superelements

"alert-service" on page 22

Subelements

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

TABLE 1–8 `alert-subscription` Subelements

Element	Required	Description
"listener-config" on page 72	only one	Configures the listener class that listens for alerts from notification emitters.
"filter-config" on page 50	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 24

Subelements

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

TABLE 1–10 `appclient-module` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	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.

TABLE 1–11 `appclient-module` Attributes *(Continued)*

Attribute	Default	Description
`location`	none	The location of the ACC module in the Application Server file system.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed to 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.

Superelements

"server" on page 98

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 110 elements to which the "web-module" on page 114 or the web modules within this "j2ee-application" on page 59 are deployed.
`lb-enabled`	`false`	(optional) If `true`, all load-balancers consider this application 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 71, "j2ee-application" on page 59, "ejb-module" on page 43, "web-module" on page 114, "connector-module" on page 34, or "appclient-module" on page 23 element.

applications

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

Superelements

"domain" on page 40

Subelements

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

TABLE 1–13 `applications` Subelements

Element	Required	Description
"lifecycle-module" on page 71	zero or more	Specifies a deployed lifecycle module.
"j2ee-application" on page 59	zero or more	Specifies a deployed Java EE application.
"ejb-module" on page 43	zero or more	Specifies a deployed EJB module.
"web-module" on page 114	zero or more	Specifies a deployed web module.
"connector-module" on page 34	zero or more	Specifies a deployed connector module.
"appclient-module" on page 23	zero or more	Specifies a deployed application client container (ACC) module.
"mbean" on page 79	zero or more	Specifies an MBean.

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 97

Subelements

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

TABLE 1–14 `audit-module` Subelements

Element	Required	Description
"property" on page 88	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`	none	Specifies the name of this audit module.
`classname`	none	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 Platform Edition 9 Developer's Guide*.

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

```
<auth-realm name="file"
     classname="com.iplanet.ias.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

"security-service" on page 97

Subelements

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

TABLE 1–16 `auth-realm` Subelements

Element	Required	Description
"property" on page 88	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 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, 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`	(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`.

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

Property	Realms	Description
`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`.

B

backend-principal

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

Superelements

"security-map" on page 96

Subelements

none

Attributes

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

TABLE 1–19 `backend-principal` Attributes

Attribute	Default	Description
`user-name`	none	Specifies the user name required by the EIS.
`password`	none	Specifies the password required by the EIS.

C

config

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

Superelements

"configs" on page 30

Subelements

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

TABLE 1–20 `config` Subelements

Element	Required	Description
"http-service" on page 55	only one	Configures the HTTP service.
"iiop-service" on page 58	only one	Configures the IIOP service.
"admin-service" on page 21	only one	Determines whether the server to which the configuration applies is an administration server.
"connector-service" on page 36	zero or one	Configures the connector service.
"web-container" on page 113	only one	Configures the web container.
"ejb-container" on page 41	only one	Configures the Enterprise JavaBeans (EJB) container.
"mdb-container" on page 80	only one	Configures the message-driven bean (MDB) container.
"jms-service" on page 68	zero or one	Configures the Java Message Service (JMS) provider.
"log-service" on page 73	only one	Configures the system logging service.
"security-service" on page 97	only one	Configures the Java EE security service.
"transaction-service" on page 107	only one	Configures the transaction service.
"monitoring-service" on page 84	only one	Configures the monitoring service.
"diagnostic-service" on page 39	zero or one	Configures the diagnostic service.
"java-config" on page 61	only one	Configures the Java Virtual Machine (JVM).
"thread-pools" on page 107	only one	Configures thread pools.
"alert-service" on page 22	zero or one	Configures the alert service.
"management-rules" on page 77	zero or one	Configures self-management rules.
"system-property" on page 104	zero or more	Specifies a system property.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–21 `config` Attributes

Attribute	Default	Description
`name`	`server-config`	Specifies the name of the configuration. For the Platform Edition, the default is the only value allowed.
`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. In the Platform Edition, there is only one configuration.

Superelements

"domain" on page 40

Subelements

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

TABLE 1–22 `configs` Subelements

Element	Required	Description
"config" on page 28	only one	Defines a configuration.

connection-pool

Defines a pool of client HTTP connections used by the "http-listener" on page 52 subelements of the parent "http-service" on page 55 element.

Superelements

"http-service" on page 55

Subelements

none

Attributes

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

TABLE 1–23 `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 52 elements.
`max-pending-count`	`4096`	(optional) Specifies the maximum number of pending connections on an "http-listener" on page 52.
`receive-buffer-size-in-bytes`	`4096`	(optional) Specifies the size of the receive buffer for all "http-listener" on page 52 elements.
`send-buffer-size-in-bytes`	`8092`	(optional) Specifies the size of the send buffer for all "http-listener" on page 52 elements.

connector-connection-pool

Defines a connector connection pool.

Superelements

"resources" on page 95

Subelements

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

TABLE 1–24 `connector-connection-pool` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"security-map" on page 96	zero or more	Maps the principal received during servlet or EJB authentication to the credentials accepted by the EIS.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–25 `connector-connection-pool` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connection pool. A "connector-resource" on page 35 element's `pool-name` attribute refers to this `name`.
`resource-adapter-name`	none	Specifies the `name` attribute of the deployed "connector-module" on page 34. 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.
`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.

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 this 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 2006Q2 Administration Guide*.

TABLE 1–26 `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.

TABLE 1–26 `connector-connection-pool` Properties *(Continued)*

Property	Default	Description
`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.
`LazyConnectionEnlistment`	`true`	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.
`LazyConnectionAssociation`	`true`	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.
`AssociateWithThread`	`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.
`MatchConnections`	`true`	If `true`, enables connection matching. You can set to `false` if connections are homogeneous.

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

connector-module

Specifies a deployed connector module.

Superelements

"applications" on page 24

Subelements

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

TABLE 1–27 `connector-module` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–28 `connector-module` Attributes

Attribute	Default	Description
`name`	`.rar` file name	The name of the connector module.
`location`	none	The location of the connector module in the 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 connector module is enabled.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed to a directory.

connector-resource

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

Superelements

"resources" on page 95

Subelements

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

TABLE 1–29 `connector-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–30 `connector-resource` Attributes

Attribute	Default	Description
`jndi-name`	none	Specifies the JNDI name for the resource.
`pool-name`	none	Specifies the `name` of the associated connector connection pool, defined in a "connector-connection-pool" on page 31 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 28

Subelements

none

Attributes

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

TABLE 1–31 `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 Application Server shutdown continues.

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 95

Subelements

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

TABLE 1–32 `custom-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–33 `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.
`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 21

Subelements

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

TABLE 1–34 `das-config` Subelements

Element	Required	Description
"property" on page 88	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 Platform Edition 9 Developer's Guide*.

TABLE 1–35 `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 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.

TABLE 1–35 `das-config` Attributes *(Continued)*

Attribute	Default	Description
`deploy-xml-validation`	`full`	(optional) Specifies the type of XML validation performed on standard and 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 20, "appclient-module" on page 23, "connector-connection-pool" on page 31, "connector-module" on page 34, "connector-resource" on page 35, "custom-resource" on page 36, "ejb-module" on page 43, "external-jndi-resource" on page 49, "j2ee-application" on page 59, "jdbc-connection-pool" on page 63, "jdbc-resource" on page 66, "lifecycle-module" on page 71, "mail-resource" on page 74, "mbean" on page 79, "persistence-manager-factory-resource" on page 86, "property" on page 88, "system-property" on page 104, "transformation-rule" on page 109, "web-module" on page 114

Subelements

none - contains data

diagnostic-service

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

Superelements

"config" on page 28

Subelements

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

TABLE 1–36 `diagnostic-service` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–37 `diagnostic-service` Attributes

Attribute	Default	Description
`compute-checksum`	`true`	(optional) If `true`, computes a checksum of binaries.
`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 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 82 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 1–38 `domain` Subelements

Element	Required	Description
"applications" on page 24	zero or one	Contains deployed Java EE applications, Java EE modules, and lifecycle modules.
"resources" on page 95	zero or one	Contains configured resources.
"configs" on page 30	only one	Contains configurations.
"servers" on page 99	only one	Contains server instances.
"system-property" on page 104	zero or more	Specifies a system property.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–39 `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 73 description for details about logs.
`locale`	operating system default	(optional) Specifies the domain's language.

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 28

Subelements

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

TABLE 1–40 `ejb-container` Subelements

Element	Required	Description
"ejb-timer-service" on page 44	zero or one	Configures the EJB timer service.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–41 `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.
`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.

TABLE 1–41 `ejb-container` Attributes *(Continued)*

Attribute	Default	Description
`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 98 element determines the location of the session store. Applies to stateful session beans.
`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-module

Specifies a deployed EJB module.

Superelements

"applications" on page 24

Subelements

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

TABLE 1–42 `ejb-module` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 115	zero or more	Configures a web service endpoint.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–43 `ejb-module` Attributes

Attribute	Default	Description
`name`	none	The name of the EJB module.
`location`	none	The location of the EJB module in the 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.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed to a directory.

ejb-timer-service

Configures the EJB timer service.

Superelements

"ejb-container" on page 41

Subelements

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

TABLE 1–44 `ejb-timer-service` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–45 `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 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 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 `Notification` mechanism.

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

Superelements

"management-rule" on page 76

Subelements

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

TABLE 1–46 `event` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–47 `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 `lifecycle`, `log`, `timer`, and `trace` types are predefined events provided by the Application Server. ■ `lifecycle` — A lifecycle event. For more information about the server life cycle, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*. ■ `log` — An event in the server log. For more information about the server log, see "log-service" on page 73. ■ `monitor` — A monitoring event, which is a change in the attribute of a monitored "mbean" on page 79. ■ `notification` — A JMX notification event. Any custom "mbean" on page 79 that can emit a notification can be a notification event. For more information about MBeans, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*. ■ `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`	depends on event	(optional) Specifies the level at which to log the event occurrence. For information about log levels, see "module-log-levels" on page 82. Applicable only if `record-event` is set to `true`. The default `level` depends on the `event` attributes and properties. For example, if the `type` attribute is set to `cluster` and the `name` property is set to `fail`, the default `level` is `SEVERE`.
`record-event`	`true`	(optional) Specifies whether the occurrence of the event is logged. If no "action" on page 19 is specified for the parent "management-rule" on page 76, 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.

TABLE 1–48 event Properties

Event Type	Property	Values	Description
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	logger	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 82.
log	level	A comma-separated list of log levels. The default is INFO	(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 82.
monitor	sourceMBean	A name attribute of a user-defined "mbean" on page 79, or a JMX ObjectName for a system mbean	Specifies the name of the monitored MBean. For more information about MBeans, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*.
monitor	attribute	An "mbean" on page 79 Attribute name	Specifies the monitored attribute of the monitored MBean. **Note** – This is *not* an attribute of the mbean element in the domain.xml file.
monitor	monitorType	CountMonitor, GaugeMonitor, StringMonitor, CountStatisticMonitor, TimeStatisticMonitor, RangeStatisticMonitor, BoundStatisticMonitor, BoundedRangeStatisticMonitor	(optional) The type of monitoring of the attribute. The default is determined by the type of the attribute to be monitored. String types default to StringMonitor, int types default to CountMonitor, and JSR 77 Statistic types default to the corresponding StatisticMonitor.
monitor	statisticAttribute	Statistic object field	Specifies the field of the Statistic object to be monitored. Required if monitorType is a StatisticMonitor.
monitor	granularity	Time interval in seconds (long int)	(optional) Specifies the granularity at which the monitoring data should be collected.
monitor	equals	A String	Specifies the value to which the attribute value is compared. Required if the monitor type is StringMonitor or the JSR 77 Statistic field being monitored is of type String.

TABLE 1–48 event Properties *(Continued)*

Event Type	Property	Values	Description
monitor	threshold	A positive number (long int)	Specifies a value above which notification occurs. Required if the monitor is of type CountMonitor or the JSR 77 Statistic field being monitored is of an int type containing a count value.
monitor	highThreshold	A positive number (long int)	Specifies the upper limit of the range within which notification occurs. Required if the monitor is of type GaugeMonitor or the JSR 77 Statistic field being monitored is of an int type containing a gauge value.
monitor	lowThreshold	A positive number (long int)	Specifies the lower limit of the range within which notification occurs. Required if the monitor is of type GaugeMonitor or the JSR 77 Statistic field being monitored is of an int type containing a gauge value.
notification	sourceMBean	name of "mbean" on page 79	Specifies a custom MBean that emits a notification. For more information about MBeans, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*.
timer	date	*mm/dd/yyyy hh:mm:ss*	Begins notification at the specified date and time.
timer	period	Time interval in milliseconds (long int)	(optional) Notifies 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 written to the server log when notification occurs.

TABLE 1–48 `event` Properties *(Continued)*

Event Type	Property	Values	Description
`trace`	`tracepoint`	A comma-separated list of trace points, or * for all trace points	Notifies at the specified trace points. Valid `tracepoint` values are as follows: `web_component_method_entry`, `web_component_method_exit`, `webservice_method_entry`, `webservice_method_exit`, `ejb_component_method_entry`, `ejb_component_method_exit`, `transaction_start`, `transaction_commit`, `transaction_rollback`, `jdbc_get_connection`, `jdbc_close_connection`, `connector_get_connection`, `connector_close_connection`.

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 95

Subelements

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

TABLE 1–49 `external-jndi-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–50 `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 Platform Edition 9 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 72.

Superelements

"alert-subscription" on page 22

Subelements

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

TABLE 1–51 `filter-config` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–52 `filter-config` Attributes

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

H

http-access-log

Defines an access log file for a "virtual-server" on page 110. The "access-log" on page 18 subelement of the virtual server's parent "http-service" on page 55 element determines the access log file's format and rotation settings.

Superelements

"virtual-server" on page 110

Subelements

none

Attributes

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

TABLE 1–53 `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 DNL lookup.

http-file-cache

Configures the HTTP file cache.

Superelements

"http-service" on page 55

Subelements

none

Attributes

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

TABLE 1–54 `http-file-cache` Attributes

Attribute	Default	Description
`globally-enabled`	`true`	(optional) If `true`, enables the file cache.
`file-caching-enabled`	`on`	(optional) If on, 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.

Superelements

"http-service" on page 55

Subelements

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

TABLE 1–55 `http-listener` Subelements

Element	Required	Description
"ssl" on page 102	zero or one	Defines Secure Socket Layer (SSL) parameters.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–56 `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.
`acceptor-threads`	`1`	(optional) Specifies the number of processors in the machine. The only legal value is `1`. To set the number of request processing threads, use the `thread-count` attribute of the "request-processing" on page 92 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 110 for this particular listener.
`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.
`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.

TABLE 1–56 `http-listener` Attributes *(Continued)*

Attribute	Default	Description
`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.

Properties

The following table describes properties for the `http-listener` element. Any of these properties can be defined as an "http-service" on page 55 property, so that it applies to all `http-listener` elements.

TABLE 1–57 `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`.
`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 55 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 55 element are both set to `true`. Overrides the `proxyHandler` property of the parent `http-service` element.
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`12000` (12 seconds)	Specifies the number of milliseconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.
`maxKeepAliveRequests`	`1000`	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.

TABLE 1–57 `http-listener` Properties *(Continued)*

Property	Default	Description
`traceEnabled`	`true`	If `true`, enables the TRACE operation. Set this property to `false` to make the Application Server less susceptible to cross-site scripting attacks.

http-service

Defines the HTTP service.

Superelements

"config" on page 28

Subelements

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

TABLE 1–58 `http-service` Subelements

Element	Required	Description
"access-log" on page 18	zero or one	Defines access log settings for each "http-access-log" on page 51 subelement of each "virtual-server" on page 110.
"http-listener" on page 52	one or more	Defines an HTTP listen socket.
"virtual-server" on page 110	one or more	Defines a virtual server.
"request-processing" on page 92	zero or one	Configures request processing threads.
"keep-alive" on page 70	zero or one	Configures keep-alive threads.
"connection-pool" on page 30	zero or one	Defines a pool of client HTTP connections.
"property" on page 88	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 52 subelements, except for `accessLogBufferSize` and `accessLogWriterInterval`, which apply to all "virtual-server" on page 110 subelements.

TABLE 1–59 `http-service` Properties

Property	Default	Description
`monitoring-cache-enabled`	`true`	If `true`, enables the monitoring cache.

TABLE 1–59 `http-service` Properties *(Continued)*

Property	Default	Description
`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.
`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 52 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 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 52 subelement can override the `proxyHandler` setting for itself.

TABLE 1–59 `http-service` Properties *(Continued)*

Property	Default	Description
`bufferSize`	`4096`	Specifies the size, in bytes, of the buffer to be provided for input streams created by HTTP listeners.
`connectionTimeout`	`12000` (12 seconds)	Specifies the number of milliseconds HTTP listeners wait, after accepting a connection, for the request URI line to be presented.
`maxKeepAliveRequests`	`1000`	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 Application Server less susceptible to cross-site scripting attacks.
`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.

I

iiop-listener

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

Superelements

"iiop-service" on page 58

Subelements

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

TABLE 1–60 `iiop-listener` Subelements

Element	Required	Description
"ssl" on page 102	zero or one	Defines SSL parameters.

TABLE 1–60 `iiop-listener` Subelements *(Continued)*

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–61 `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 28

Subelements

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

TABLE 1–62 `iiop-service` Subelements

Element	Required	Description
"orb" on page 85	only one	Configures the ORB.
"ssl-client-config" on page 103	zero or one	Defines SSL parameters for the ORB.
"iiop-listener" on page 57	zero or more	Defines an IIOP listen socket.

Attributes

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

TABLE 1–63 `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.

J

j2ee-application

Specifies a deployed Java EE application.

Superelements

"applications" on page 24

Subelements

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

TABLE 1–64 `j2ee-application` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 115	zero or more	Configures a web service endpoint.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–65 `j2ee-application` Attributes

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

TABLE 1–65 `j2ee-application` 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 application 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.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed to 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 97

Subelements

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

TABLE 1–66 `jacc-provider` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–67 `jacc-provider` Attributes

Attribute	Default	Description
`name`	`default`	Specifies the name of the JACC provider.
`policy-provider`	none	Corresponds to and can be overridden by the system property `javax.security.jacc.policy.provider`.
`policy-configuration-factory-provider`	none	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 28

Subelements

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

TABLE 1–68 `java-config` Subelements

Element	Required	Description
"profiler" on page 87	zero or one	Configures a profiler for use with the Application Server.
"jvm-options" on page 70	zero or more	Contains JVM command line options.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–69 `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 1–69 `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 Platform Edition 9 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 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 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.

TABLE 1–69 `java-config` Attributes *(Continued)*

Attribute	Default	Description
`env-classpath-ignored`	`true`	(optional) If `false`, the `CLASSPATH` environment variable is read and appended to the 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 95

Subelements

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

TABLE 1–70 `jdbc-connection-pool` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–71 `jdbc-connection-pool` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the connection pool. A "jdbc-resource" on page 66 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 `java.sql.DataSource`, `java.sql.XADataSource` , `javax.sql.ConnectionPoolDatasource`, or a combination.

TABLE 1–71 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`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.
`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.

TABLE 1–71 `jdbc-connection-pool` Attributes *(Continued)*

Attribute	Default	Description
`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.
`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.

Properties

Most JDBC 3.0 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 Application Server, some properties might be necessary for most databases. For details, see the JDBC 3.0 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.

TABLE 1–72 `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`	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. Default is `true`.
`LazyConnection Association`	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. Default is `true`.
`AssociateWithThread`	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. Default is `false`.
`MatchConnections`	If `true`, enables connection matching. You can set to `false` if connections are homogeneous. Default is `true`.

jdbc-resource

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

Superelements

"resources" on page 95

Subelements

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

TABLE 1–73 `jdbc-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–74 `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 63.
`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-host

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

Superelements

"jms-service" on page 68

Subelements

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

TABLE 1–75 `jms-host` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–76 `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 Application Server.

Superelements

"config" on page 28

Subelements

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

TABLE 1–77 `jms-service` Subelements

Element	Required	Description
"jms-host" on page 67	zero or more	Specifies a host.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–78 `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.

TABLE 1–78 `jms-service` Attributes *(Continued)*

Attribute	Default	Description
`type`	`EMBEDDED`	(optional) Specifies the type of JMS service: ■ `EMBEDDED` means the JMS provider is started in the same JVM as the Application Server. This is the default for the Domain Administration Server (DAS). ■ `LOCAL` means the JMS provider is started along with the Application Server. ■ `REMOTE` means the JMS provider is remote and is not started by the 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 67. If `type` is set to `LOCAL`, this `jms-host` is automatically started at Application Server startup.
`reconnect-interval-in-seconds`	`5`	(optional) Specifies the interval between reconnect attempts.
`reconnect-attempts`	`3`	(optional) Specifies the number of reconnect attempts.
`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 1–79 `jms-service` Properties

Property	Default	Description
`instance-name`	`imqbroker`	Specifies the full Sun Java System Message Queue broker instance name.

TABLE 1–79 `jms-service` Properties *(Continued)*

Property	Default	Description
`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`.

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 61, "profiler" on page 87

Subelements

none - contains data

K

keep-alive

Configures keep-alive threads.

Superelements

"http-service" on page 55

Subelements

none

Attributes

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

TABLE 1–80 `keep-alive` Attributes

Attribute	Default	Description
`thread-count`	`1`	(optional) Specifies the number of keep-alive threads.
`max-connections`	`256`	(optional) Specifies the maximum number of keep-alive connections.
`timeout-in-seconds`	`60`	(optional) Specifies the maximum time for which a keep alive connection is kept open.

L

lifecycle-module

Specifies a deployed lifecycle module. For more information about lifecycle modules, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*.

Superelements

"applications" on page 24

Subelements

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

TABLE 1–81 `lifecycle-module` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–82 `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.

TABLE 1–82 `lifecycle-module` Attributes *(Continued)*

Attribute	Default	Description
`classpath`	value of `application-root` attribute of "server" on page 98 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.
`is-failure-fatal`	`false`	(optional) Determines whether the server is shut down if the lifecycle module fails.
`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 22

Subelements

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

TABLE 1–83 `listener-config` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–84 `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 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 Application Server, but custom emitters can be used.

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 110.
- The *access log* file stores HTTP access messages from the default virtual server. The default name is `access.log`. See "access-log" on page 18 and "http-access-log" on page 51.
- 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 107.

Superelements

"config" on page 28

Subelements

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

TABLE 1–85 `log-service` Subelements

Element	Required	Description
"module-log-levels" on page 82	zero or one	Specifies log levels.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–86 `log-service` Attributes

Attribute	Default	Description
`file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 40 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 40 element. A relative path is relative to the `log-root` attribute of the "domain" on page 40 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 95

Subelements

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

TABLE 1–87 `mail-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–88 `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 1–88 `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 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");
```

management-rule

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

Superelements

"management-rules" on page 77

Subelements

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

TABLE 1–89 `management-rule` Subelements

Element	Required	Description
"description" on page 39	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 73 and "module-log-levels" on page 82.
"event" on page 45	only one	Defines the event that triggers the action associated with a management rule.
"action" on page 19	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 1–90 `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 Application Server.

Superelements

"config" on page 28

Subelements

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

TABLE 1–91 `management-rules` Subelements

Element	Required	Description
"management-rule" on page 76	zero or more	Specifies a management rule.

Attributes

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

TABLE 1–92 `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 100

Subelements

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

TABLE 1–93 `manager-properties` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–94 `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.
`reap-interval-in-seconds`	`60`	(optional) Specifies the time between checks for expired sessions. 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.

TABLE 1–94 `manager-properties` Attributes *(Continued)*

Attribute	Default	Description
`session-id-generator-classname`	internal class generator	(optional) Not implemented. Do not use.

mbean

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

Superelements

"applications" on page 24

Subelements

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

TABLE 1–95 `mbean` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	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 1–96 `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.

TABLE 1–96 `mbean` Attributes *(Continued)*

Attribute	Default	Description
`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 28

Subelements

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

TABLE 1–97 `mdb-container` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–98 `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 1–99 `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 97

Subelements

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

TABLE 1–100 `message-security-config` Subelements

Element	Required	Description
"provider-config" on page 89	one or more	Specifies a configuration for one message security provider.

Attributes

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

TABLE 1–101 `message-security-config` Attributes

Attribute	Default	Description
`auth-layer`	none	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 73

Subelements

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

TABLE 1–102 `module-log-levels` Subelements

Element	Required	Description
"property" on page 88	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 Application Server system loggers.

TABLE 1–103 `module-log-levels` Attributes

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

TABLE 1–103 `module-log-levels` Attributes *(Continued)*

Attribute	Default	Description
`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.
`self-management`	`INFO`	(optional) Specifies the level of messages logged by the self-management (management rules) subsystem.
`management-event`	`INFO`	(optional) Specifies the level of messages logged by the self-management event 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 84

Subelements

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

TABLE 1–104 `module-monitoring-levels` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–105 `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.

monitoring-service

Configures the monitoring service.

Superelements

"config" on page 28

Subelements

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

TABLE 1–106 `monitoring-service` Subelements

Element	Required	Description
"module-monitoring-levels" on page 83	zero or one	Controls the level of monitoring of server subsystems.
"property" on page 88	zero or more	Specifies a property or a variable.

O

orb

Configures the ORB.

To enable SSL for outbound connections, include an "ssl-client-config" on page 103 subelement in the parent `iiop-service` element.

Superelements

"iiop-service" on page 58

Subelements

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

TABLE 1–107 `orb` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–108 `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 106 elements used by the ORB.
`message-fragment-size`	`1024`	(optional) GIOPv1.2 messages larger than this number of bytes are fragmented.

TABLE 1–108 `orb` Attributes *(Continued)*

Attribute	Default	Description
`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 66 element instead.

Superelements

"resources" on page 95

Subelements

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

TABLE 1–109 `persistence-manager-factory-resource` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–110 `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.

TABLE 1–110 `persistence-manager-factory-resource` Attributes *(Continued)*

Attribute	Default	Description
`jdbc-resource -jndi-name`	none	Specifies the "jdbc-resource" on page 66 from which database connections are obtained. Must be the `jndi-name` of an existing `jdbc-resource`.
`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 or EJB client.

Superelements

"security-map" on page 96

Subelements

none - contains data

profiler

Configures a profiler for use with the Application Server. For more information about profilers, see the *Sun Java System Application Server Platform Edition 9 Developer's Guide*.

Superelements

"java-config" on page 61

Subelements

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

TABLE 1–111 `profiler` Subelements

Element	Required	Description
"jvm-options" on page 70	zero or more	Contains profiler-specific JVM command line options.
"property" on page 88	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 1–112 `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 Application Server
- Needed by a system or object that the 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 20, "admin-service" on page 21, "alert-service" on page 22, "appclient-module" on page 23, "audit-module" on page 25, "auth-realm" on page 26, "config" on page 28, "connector-connection-pool" on page 31, "connector-module" on page 34, "connector-resource" on page 35, "custom-resource" on page 36, "das-config" on page 37, "diagnostic-service" on page 39, "domain" on page 40, "ejb-container" on page 41, "ejb-module" on page 43, "ejb-timer-service" on page 44, "external-jndi-resource" on page 49, "filter-config" on page 50, "http-listener" on page 52, "http-service" on page 55, "iiop-listener" on page 57, "j2ee-application" on page 59, "jacc-provider" on page 60, "java-config" on page 61, "jdbc-connection-pool" on page 63, "jdbc-resource" on page 66, "jms-host" on page 67, "jms-service" on page 68, "lifecycle-module" on page 71, "listener-config" on page 72, "log-service" on page 73, "mail-resource" on page 74, "manager-properties" on page 78, "mbean" on page 79, "mdb-container" on page 80, "module-log-levels" on page 82, "module-monitoring-levels" on page 83, "monitoring-service" on page 84, "orb" on page 85, "persistence-manager-factory-resource" on page 86, "profiler" on page 87, "provider-config" on page 89, "resource-adapter-config" on page 93, "security-service" on page 97, "server" on page 98, "session-properties" on page 101, "store-properties" on page 103, "transaction-service" on page 107, "virtual-server" on page 110, "web-container" on page 113, "web-module" on page 114

Subelements

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

TABLE 1–113 `property` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.

Attributes

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

TABLE 1–114 `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 81

Subelements

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

TABLE 1–115 `provider-config` Subelements

Element	Required	Description
"request-policy" on page 92	zero or one	Defines the authentication policy requirements of the authentication provider's request processing.
"response-policy" on page 96	zero or one	Defines the authentication policy requirements of the authentication provider's response processing.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–116 `provider-config` Attributes

Attribute	Default	Description
`provider-id`	none	Specifies a unique identifier for this `provider-config` element.
`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 1–117 `provider-config` Properties

Property	Default	Description
`security.config`	none	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-2.0.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.

R

registry-location

Specifies the registry where web service endpoint artifacts are published.

Superelements

"web-service-endpoint" on page 115

Subelements

none

Attributes

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

TABLE 1–118 `registry-location` Attributes

Attribute	Default	Description
`connector-resource-jndi-name`	none	Specifies the `jndi-name` of the "connector-resource" on page 35 used as the registry.

request-policy

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

Superelements

"provider-config" on page 89

Subelements

none

Attributes

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

TABLE 1–119 `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 55

Subelements

none

Attributes

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

TABLE 1–120 `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-in-bytes`	`4096`	(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 95

Subelements

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

TABLE 1–121 `resource-adapter-config` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–122 `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 106 element.

TABLE 1–122 `resource-adapter-config` 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.
`resource-adapter-name`	none	Specifies the `name` attribute of a deployed "connector-module" on page 34. 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.

Superelements

"server" on page 98

Subelements

none

Attributes

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

TABLE 1–123 `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 36, "external-jndi-resource" on page 49, "jdbc-resource" on page 66, "mail-resource" on page 74, "persistence-manager-factory-resource" on page 86, "admin-object-resource" on page 20 "resource-adapter-config" on page 93, "jdbc-connection-pool" on page 63, or "connector-connection-pool" on page 31 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 Application Server application should begin with the string `java:comp/env`.

Superelements

"domain" on page 40

Subelements

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

TABLE 1–124 `resources` Subelements

Element	Required	Description
"custom-resource" on page 36	zero or more	Defines a custom resource.
"external-jndi-resource" on page 49	zero or more	Defines a resource that resides in an external JNDI repository.
"jdbc-resource" on page 66	zero or more	Defines a JDBC (Java Database Connectivity) resource.
"mail-resource" on page 74	zero or more	Defines a JavaMail resource.
"persistence-manager-factory-resource" on page 86	zero or more	Defines a persistence manager factory resource for CMP. Deprecated. Use a "jdbc-resource" on page 66 element instead.
"admin-object-resource" on page 20	zero or more	Defines an administered object for an inbound resource adapter.
"connector-resource" on page 35	zero or more	Defines a connector (resource adapter) resource.
"resource-adapter-config" on page 93	zero or more	Defines a resource adapter configuration.
"jdbc-connection-pool" on page 63	zero or more	Defines the properties that are required for creating a JDBC connection pool.
"connector-connection-pool" on page 31	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 89

Subelements

none

Attributes

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

TABLE 1–125 `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 31

Subelements

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

TABLE 1–126 `security-map` Subelements

Element	Required	Description
"principal" on page 87	one or more	Contains the principal of the servlet or EJB client.
"user-group" on page 110	one or more	Contains the group to which the principal belongs.
"backend-principal" on page 28	only one	Specifies the user name and password required by the EIS.

Attributes

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

TABLE 1–127 `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 102. For connector module security, see "security-map" on page 96.

Superelements

"config" on page 28

Subelements

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

TABLE 1–128 `security-service` Subelements

Element	Required	Description
"auth-realm" on page 26	one or more	Defines a realm for authentication.
"jacc-provider" on page 60	one or more	Specifies a Java Authorization Contract for Containers (JACC) provider for pluggable authorization.
"audit-module" on page 25	zero or more	Specifies an optional plug-in module that implements audit capabilities.
"message-security-config" on page 81	zero or more	Specifies configurations for message security providers.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–129 `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.
`anonymous-role`	`ANYONE`	(optional) Used as the name for default, or anonymous, role. The anonymous role is always assigned to all principals. This role value can be used in Java EE deployment descriptors to grant access to anyone.
`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 60 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.

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 99

Subelements

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

TABLE 1–130 `server` Subelements

Element	Required	Description
"application-ref" on page 24	zero or more	References an application or module deployed to the server instance.
"resource-ref" on page 94	zero or more	References a resource deployed to the server instance.
"system-property" on page 104	zero or more	Specifies a system property.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–131 `server` Attributes

Attribute	Default	Description
`name`	none	Specifies the name of the server instance.
`config-ref`	default "config" on page 28 element's `name`, `server-config`	(optional) References the `name` of the "config" on page 28 used by the server instance. For the Platform Edition, the default is the only value allowed.

servers

Contains server instances. In the Platform Edition, there is only one server instance.

Superelements

"domain" on page 40

Subelements

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

TABLE 1–132 `servers` Subelements

Element	Required	Description
"server" on page 98	only one	Defines a server instance.

session-config

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

Superelements

"web-container" on page 113

Subelements

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

TABLE 1–133 `session-config` Subelements

Element	Required	Description
"session-manager" on page 100	zero or one	Specifies session manager configuration information.
"session-properties" on page 101	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 100

Subelements

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

TABLE 1–134 `session-manager` Subelements

Element	Required	Description
"manager-properties" on page 78	zero or one	Specifies session manager properties.
"store-properties" on page 103	zero or one	Specifies session persistence (storage) properties.

session-properties

Specifies session properties.

Superelements

"session-config" on page 100

Subelements

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

TABLE 1–135 `session-properties` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–136 `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 module. If set to `0` or less, sessions in this web module never expire. If a `session-timeout` element is specified in the `web.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` is specified in minutes, not seconds.

Properties

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

TABLE 1–137 `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 module's session ID.

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`.

In Platform Edition, SSL is globally disabled.

Superelements

"http-listener" on page 52, "iiop-listener" on page 57, "ssl-client-config" on page 103

Subelements

none

Attributes

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

TABLE 1–138 `ssl` Attributes

Attribute	Default	Description
`cert-nickname`	none	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 110, the server tries SSL3 encryption first. If that fails, the server tries SSL2 encryption.

TABLE 1–138 `ssl` Attributes *(Continued)*

Attribute	Default	Description
`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 110, 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.
`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 58

Subelements

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

TABLE 1–139 `ssl-client-config` Subelements

Element	Required	Description
"ssl" on page 102	only one	Defines SSL parameters.

store-properties

Specifies session persistence (storage) properties.

Superelements

"session-manager" on page 100

Subelements

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

TABLE 1–140 `store-properties` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

TABLE 1–141 `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.
`reap-interval-in-seconds`	`60`	(optional) Not implemented. Use the `reap-interval-in-seconds` attribute of the "manager-properties" on page 78 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 40, "server" on page 98, or "config" on page 28. A value set at a higher level can be overridden at a lower level. Some system properties are predefined; see "system-property" on page 104. 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

"config" on page 28, "domain" on page 40, "server" on page 98

Subelements

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

TABLE 1–142 `system-property` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.

Attributes

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

TABLE 1–143 `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 1–144 Predefined System Properties

Property	Default	Description
`com.sun.aas.installRoot`	depends on operating system	Specifies the directory where the 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).

TABLE 1–144 Predefined System Properties *(Continued)*

Property	Default	Description
`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 28 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.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 107

Subelements

none

Attributes

TABLE 1–145 `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.

TABLE 1–145 `thread-pool` Attributes *(Continued)*

Attribute	Default	Description
`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 28

Subelements

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

TABLE 1–146 `thread-pools` Subelements

Element	Required	Description
"thread-pool" on page 106	one or more	Defines a thread pool.

transaction-service

Configures the Java Transaction Service (JTS).

Superelements

"config" on page 28

Subelements

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

TABLE 1–147 `transaction-service` Subelements

Element	Required	Description
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–148 `transaction-service` Attributes

Attribute	Default	Description
`automatic-recovery`	`false`	(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 40 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 40 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`	`2048`	(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 (for example, `4096`) results in a larger transaction log file, but fewer keypoint operations and potentially better performance. A smaller value (for example, `100`) 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 1–149 `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.

TABLE 1–149 `transaction-service` Properties *(Continued)*

Property	Default	Description
`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.

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 115

Subelements

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

TABLE 1–150 `transformation-rule` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.

Attributes

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

TABLE 1–151 `transformation-rule` Attributes

Attribute	Default	Description
`name`	none	The name of the rule.
`enabled`	`true`	(optional) Determines whether the rule is enabled.

TABLE 1–151 `transformation-rule` Attributes *(Continued)*

Attribute	Default	Description
`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.

U

user-group

Contains the group to which the principal belongs.

Superelements

"security-map" on page 96

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 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 52 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 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 55

Subelements

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

TABLE 1–152 `virtual-server` Subelements

Element	Required	Description
"http-access-log" on page 51	zero or one	Defines an access log file.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–153 `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 52 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`.

TABLE 1–153 `virtual-server` Attributes *(Continued)*

Attribute	Default	Description
`default-web-module`	system default web module	(optional) References the `name` attribute of the default "web-module" on page 114 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 24 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.
`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`.
`log-file`	`server.log` in the directory specified by the `log-root` attribute of the "domain" on page 40 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 73 description for details about logs.

Properties

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

TABLE 1–154 `virtual-server` Properties

Property	Default	Description
`sso-enabled`	`true`	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.
`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.
`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`. To set this property for all virtual servers, set it as a property of the parent "http-service" on page 55 element.

TABLE 1–154 `virtual-server` Properties *(Continued)*

Property	Default	Description
`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 55 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.

W

web-container

Configures the web container.

Superelements

"config" on page 28

Subelements

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

TABLE 1–155 `web-container` Subelements

Element	Required	Description
"session-config" on page 100	zero or one	Specifies session configuration information for the web container.
"property" on page 88	zero or more	Specifies a property or a variable.

Properties

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

TABLE 1–156 `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-module

Specifies a deployed web module.

Superelements

"applications" on page 24

Subelements

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

TABLE 1–157 `web-module` Subelements

Element	Required	Description
"description" on page 39	zero or one	Contains a text description of this element.
"web-service-endpoint" on page 115	zero or more	Configures a web service endpoint.
"property" on page 88	zero or more	Specifies a property or a variable.

Attributes

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

TABLE 1–158 `web-module` Attributes

Attribute	Default	Description
`name`	none	The name of the web module.
`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.
`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.
`directory-deployed`	`false`	(optional) Specifies whether the application has been deployed to 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 43, "j2ee-application" on page 59, "web-module" on page 114

Subelements

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

TABLE 1–159 `web-service-endpoint` Subelements

Element	Required	Description
"registry-location" on page 91	zero or more	Specifies the registry where web service endpoint artifacts are published.

TABLE 1–159 `web-service-endpoint` Subelements *(Continued)*

Element	Required	Description
"transformation-rule" on page 109	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 1–160 `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 83.
`max-history-size`	`25`	(optional) Specifies the maximum number of monitoring records stored for this endpoint.
`jbi-enabled`	`true`	(optional) Determines whether the visibility of this endpoint as a Java Business Integration service is enabled or disabled.

Index

D

E

K

L

M

N

O

P

Q

R

S

T

U

V

W

X