Sun ONE Application Server 7 man pages
Contents

Preface 7

Application Server Utility 15
appclient(1AS) 16
asadmin(1AS) 18
asadmin-add-resources(1AS) 39
asadmin-create-acl(1AS) 40
asadmin-create-authdb(1AS) 41
asadmin-create-auth-realml(1AS) 43
asadmin-create-custom-resource(1AS) 44
asadmin-create-domain(1AS) 46
asadmin-create-file-user(1AS) 47
asadmin-create-http-listener(1AS) 49
asadmin-create-http-qos(1AS) 51
asadmin-create-iiop-listener(1AS) 53
asadmin-create-instance(1AS) 55
asadmin-create-javamail-resource(1AS) 57
asadmin-create-jdbc-connection-pool(1AS) 59
asadmin-create-jdbc-resource(1AS) 62
asadmin-create-jmsdest(1AS) 64
asadmin-create-jms-resource(1AS) 65
asadmin-create-jndi-resource(1AS) 67
asadmin-create-jvm-options(1AS) 69
asadmin-create-lifecycle-module(1AS) 71
asadmin-create-mime(1AS) 73
asadmin-create-persistence-resource(1AS) 74
asadmin-create-profiler(1AS)  76
asadmin-create-ssl(1AS)  78
asadmin-create-virtual-server(1AS)  80
asadmin-delete-acl(1AS)  82
asadmin-delete-authdb(1AS)  83
asadmin-delete-auth-realm(1AS)  84
asadmin-delete-custom-resource(1AS)  85
asadmin-delete-domain(1AS)  86
asadmin-delete-file-user(1AS)  87
asadmin-delete-http-listener(1AS)  88
asadmin-delete-http-qos(1AS)  89
asadmin-delete-iiop-listener(1AS)  90
asadmin-delete-instance(1AS)  91
asadmin-delete-javamail-resource(1AS)  93
asadmin-delete-jdbc-connection-pool(1AS)  94
asadmin-delete-jdbc-resource(1AS)  95
asadmin-delete-jmsdest(1AS)  96
asadmin-delete-jms-resource(1AS)  97
asadmin-delete-jndi-resource(1AS)  98
asadmin-delete-jvm-options(1AS)  99
asadmin-delete-lifecycle-module(1AS)  101
asadmin-delete-mime(1AS)  102
asadmin-delete-persistence-resource(1AS)  103
asadmin-delete-profiler(1AS)  104
asadmin-delete-ssl(1AS)  105
asadmin-delete-virtual-server(1AS)  106
asadmin-deploy(1AS)  107
asadmin-deploydir(1AS)  109
asadmin-disable(1AS)  111
asadmin-display-license(1AS)  112
asadmin-enable(1AS)  113
asadmin-export(1AS)  114
asadmin-get(1AS)  115
asadmin-help(1AS)  117
asadmin-install-license(1AS)  120
asadmin-jms-ping(1AS)  121
asadmin-list(1AS)  122
asadmin-list-acls(1AS)  124
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>asadmin-version(1AS)</td>
<td>171</td>
</tr>
<tr>
<td>asant(1AS)</td>
<td>172</td>
</tr>
<tr>
<td>capture-schema(1AS)</td>
<td>174</td>
</tr>
<tr>
<td>flexanlg(1AS)</td>
<td>175</td>
</tr>
<tr>
<td>htpasswd(1AS)</td>
<td>177</td>
</tr>
<tr>
<td>jspc(1AS)</td>
<td>178</td>
</tr>
<tr>
<td>package-appclient(1AS)</td>
<td>180</td>
</tr>
<tr>
<td>verifier(1AS)</td>
<td>182</td>
</tr>
<tr>
<td>wscompile(1AS)</td>
<td>183</td>
</tr>
<tr>
<td>wsdeploy(1AS)</td>
<td>187</td>
</tr>
</tbody>
</table>

**Index**

191
Preface

Product Line Overview

Sun ONE Application Server 7 is a J2EE 1.3 specification-compatible application server which also supports emerging Java Web Services standards as well as standard HTTP server programming facilities. Three editions of the application server are offered to suit a variety of needs for both production and development environments:

- “Platform Edition” on page 7
- “Standard Edition” on page 8
- “Enterprise Edition” on page 8

Platform Edition

Platform Edition forms the core of the Sun ONE Application Server 7 product line. This free-for-production-use product offers a high-performance, small-footprint J2EE 1.3 specification-compatible runtime environment that is ideally suited for basic operational deployments, as well as for embedding in third-party applications. Web-services ready, the Platform Edition includes built-in technologies proven by the Sun ONE Web Server and Sun ONE Message Queue products.

Platform Edition deployments are limited to single application server instances (i.e. single virtual machines for the Java platform (“Java virtual machine” or “JVM™”)). Multi-tier deployment topologies are supported by the Platform edition, but the web server tier proxy does not perform load balancing. In Platform Edition, administrative utilities are limited to local clients only.

Platform Edition is integrated into Solaris 9.
Standard Edition

This is the edition that is the focus of this Getting Started Guide. The Standard Edition layers enhanced, remote-management capabilities on top of the Platform Edition. Enhanced management capabilities, remote command-line, and web-based administration are all included as part of the Standard Edition. This edition also includes the ability to partition web application traffic through a web server tier proxy. Standard Edition supports configuration of multiple application server instances (JVMs) per machine.

Enterprise Edition

Enterprise Edition enhances the core application server platform with greater high availability, load balancing and clustering capabilities suited for the most demanding J2EE-based application deployments. The management capabilities of the Standard Edition are extended in Enterprise Edition to account for multi-instance and multi-machine deployments.

Clustering support includes easy-to-configure groups of cloned application server instances to which client requests can be load balanced. Both external load balancers and load balancing web tier-based proxies are supported by this edition. HTTP session, stateful session bean instance and Java Message Service (JMS) resource failover are included in the Enterprise Edition. The patented "Always On" highly available database technology forms the basis for the HA persistence store in the Enterprise Edition.

For more product information, see the Sun ONE Application Server page on the Sun Microsystems web site.

Using the Documentation

The Sun ONE Application Server manuals are available as online files in Portable Document Format (PDF) and Hypertext Markup Language (HTML) formats, at:

http://docs.sun.com/

The following table lists tasks and concepts described in the Sun ONE Application Server manuals. The left column lists the tasks and concepts, and the right column lists the corresponding manuals.
<table>
<thead>
<tr>
<th>For information about</th>
<th>See the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>Late-breaking information about the software and the documentation</td>
<td>Release Notes</td>
</tr>
<tr>
<td>Supported platforms and environments</td>
<td>Platform Summary</td>
</tr>
<tr>
<td>Introduction to the application server, including new features, evaluation installation information, and architectural overview.</td>
<td>Getting Started Guide</td>
</tr>
<tr>
<td>Installing Sun ONE Application Server and its various components (sample applications, Administration interface, Sun ONE Message Queue).</td>
<td>Installation Guide</td>
</tr>
<tr>
<td>Creating and implementing J2EE applications that follow the open Java standards model on the Sun ONE Application Server 7. Includes general information about application design, developer tools, security, assembly, deployment, debugging, and creating lifecycle modules.</td>
<td>Developer’s Guide</td>
</tr>
<tr>
<td>Creating and implementing J2EE applications that follow the open Java standards model for web applications on the Sun ONE Application Server 7. Discusses web application programming concepts and tasks, and provides sample code, implementation tips, and reference material.</td>
<td>Developer’s Guide to Web Applications</td>
</tr>
<tr>
<td>Creating and implementing J2EE applications that follow the open Java standards model for enterprise beans on the Sun ONE Application Server 7. Discusses EJB programming concepts and tasks, and provides sample code, implementation tips, and reference material.</td>
<td>Developer’s Guide to Enterprise JavaBeans Technology</td>
</tr>
<tr>
<td>Creating clients that access J2EE applications on the Sun ONE Application Server 7</td>
<td>Developer’s Guide to Clients</td>
</tr>
<tr>
<td>Creating web services</td>
<td>Developer’s Guide to Web Services</td>
</tr>
<tr>
<td>J2EE features such as JDBC, JNDI, JTS, JMS, JavaMail, resources, and connectors</td>
<td>Developer’s Guide to J2EE Features and Services</td>
</tr>
<tr>
<td>Creating custom NSAPI plugins</td>
<td>Developer’s Guide to NSAPI</td>
</tr>
</tbody>
</table>
TABLE P–1 Sun ONE Application Server Documentation Roadmap (Continued)

<table>
<thead>
<tr>
<th>Performing the following administration tasks:</th>
<th>See the following</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ Using the Administration interface and the command line interface</td>
<td>Administrator’s Guide</td>
</tr>
<tr>
<td>■ Configuring server preferences</td>
<td></td>
</tr>
<tr>
<td>■ Using administrative domains</td>
<td></td>
</tr>
<tr>
<td>■ Using server instances</td>
<td></td>
</tr>
<tr>
<td>■ Monitoring and logging server activity</td>
<td></td>
</tr>
<tr>
<td>■ Configuring the web server plugin</td>
<td></td>
</tr>
<tr>
<td>■ Configuring the Java Messaging Service</td>
<td></td>
</tr>
<tr>
<td>■ Using J2EE features</td>
<td></td>
</tr>
<tr>
<td>■ Configuring support for CORBA-based clients</td>
<td></td>
</tr>
<tr>
<td>■ Configuring database connectivity</td>
<td></td>
</tr>
<tr>
<td>■ Configuring transaction management</td>
<td></td>
</tr>
<tr>
<td>■ Configuring the web container</td>
<td></td>
</tr>
<tr>
<td>■ Deploying applications</td>
<td></td>
</tr>
<tr>
<td>■ Managing virtual servers</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Editing server configuration files</th>
<th>Administrator’s Configuration File Reference</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Configuring and administering security for the Sun ONE Application Server 7 operational environment. Includes information on general security, certificates, and SSL/TLS encryption. HTTP server-based security is also addressed.</th>
<th>Administrator’s Guide to Security</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Configuring and administering service provider implementation for J2EE CA connectors for the Sun ONE Application Server 7. Includes information about the Administration Tool, DTDs and provides sample XML files.</th>
<th>J2EE CA Service Provider Implementation Administrator’s Guide</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Migrating your applications to the new Sun ONE Application Server 7 programming model from the Netscape Application Server version 2.1, including a sample migration of an Online Bank application provided with Sun ONE Application Server</th>
<th>Migrating and Redploying Server Applications Guide</th>
</tr>
</thead>
</table>

|-----------------------------|-----------------------------------------------------------------------------------------------|

Documentation Conventions

This section describes the types of conventions used throughout this guide:

■ “General Conventions” on page 11
“Conventions Referring to Directories” on page 12

General Conventions
The following general conventions are used in this guide:

- **File and directory paths** are given in UNIX® format (with forward slashes separating directory names). For Windows versions, the directory paths are the same, except that backslashes are used to separate directories.

- **URLs** are given in the format:
  
  http://server.domain/path/file.html
  
  In these URLs, server is the server name where applications are run; domain is your Internet domain name; path is the server’s directory structure; and file is an individual filename. Italic items in URLs are placeholders.

- **Font conventions** include:
  
  - The monospace font is used for sample code and code listings, API and language elements (such as function names and class names), file names, pathnames, directory names, and HTML tags.
  - *Italic* type is used for code variables.
  - *Italic* type is also used for book titles, emphasis, variables and placeholders, and words used in the literal sense.
  - **Bold** type is used as either a paragraph lead-in or to indicate words used in the literal sense.

- **Installation root directories** for most platforms are indicated by *install_dir* in this document. Exceptions are noted in “Conventions Referring to Directories” on page 12.

  By default, the location of *install_dir* on most platforms is:
  
  - Solaris 8 non-package-based Evaluation installations:
    
    user’s home directory/sun/appserver7
  
  - Solaris unbundled, non-evaluation installations:
    
    /opt/SUNWappserver7
  
  - Windows, all installations:
    
    C:\Sun\AppServer7
  
  For the platforms listed above, *default_config_dir* and *install_config_dir* are identical to *install_dir*. See “Conventions Referring to Directories” on page 12 for exceptions and additional information.

- **Instance root directories** are indicated by *instance_dir* in this document, which is an abbreviation for the following:

  *default_config_dir*/domaina/domain/instance
UNIX-specific descriptions throughout this manual apply to the Linux operating system as well, except where Linux is specifically mentioned.

Conventions Referring to Directories

By default, when using the Solaris 8 and 9 package-based installation and the Solaris 9 bundled installation, the application server files are spread across several root directories. These directories are described in this section.

- **For Solaris 9 bundled installations**, this guide uses the following document conventions to correspond to the various default installation directories provided:
  - `install_dir` refers to `/usr/appserver/`, which contains the static portion of the installation image. All utilities, executables, and libraries that make up the application server reside in this location.
  - `default_config_dir` refers to `/var/appserver/domains`, which is the default location for any domains that are created.
  - `install_config_dir` refers to `/etc/appserver/config`, which contains installation-wide configuration information such as licenses and the master list of administrative domains configured for this installation.

- **For Solaris 8 and 9 package-based, non-evaluation, unbundled installations**, this guide uses the following document conventions to correspond to the various default installation directories provided:
  - `install_dir` refers to `/opt/SUNWappserver7`, which contains the static portion of the installation image. All utilities, executables, and libraries that make up the application server reside in this location.
  - `default_config_dir` refers to `/var/opt/SUNWappserver7/domains`, which is the default location for any domains that are created.
  - `install_config_dir` refers to `/etc/opt/SUNWappserver7/config`, which contains installation-wide configuration information such as licenses and the master list of administrative domains configured for this installation.

Product Support

If you have problems with your system, contact customer support using one of the following mechanisms:

- The online support web site at:  
- The telephone dispatch number associated with your maintenance contract
Please have the following information available prior to contacting support. This helps to ensure that our support staff can best assist you in resolving problems:

- Description of the problem, including the situation where the problem occurs and its impact on your operation
- Machine type, operating system version, and product version, including any patches and other software that might be affecting the problem
- Detailed steps on the methods you have used to reproduce the problem
- Any error logs or core dumps
Application Server Utility
**NAME**

appclient – launches the Application Client Container and invokes the client application packaged in the application JAR file

**SYNOPSIS**

```bash
appclient -client client_application_jar
    [-mainclass client_application_main_classname] [-name display_name]
    [-xml sun-acc.xml file] [-textauth] [app-args]
```

**DESCRIPTION**

Use the `appclient` command to launch the application client container and invoke a client application that is packaged in an application JAR file.

The application client container is a set of java classes, libraries and other files that are required to execute a first-tier application client program on a Java Virtual Machine (JVM). The application client container communicates with the Application Server using RMI-IIOP.

**OPTIONS**

- `-client` required; the name and location for the client application jar file.
- `-mainclass` optional; the full classname of the main client application `main()` method that will be invoked by the Application Client Container. Used for a single client application.
- `-name` optional; the display name for the client application. Used for multiple client applications.
- `-xml` optional if using the default domain and instance, otherwise it is required; identifies the name and location of the client configuration XML file. If not specified, defaults to the value of `$AS_ACC_CONFIG` identified in `asenv.conf` file.
- `-textauth` optional; used to specify using text format authentication when authentication is needed.
- `app-args` optional; represents a list of arguments, separated by spaces, passed to the clients `main()` method.

**EXAMPLES**

**EXAMPLE 1** Using the `appclient` command

```bash
appclient -client sunoneappserv/bin/myclientapp.jar
    -mainclass com.sun.test.TestAppClient -xml sun-acc.xml scott sample
```

Where: `sunoneappserv/bin/myclientapp.jar` is the full path for the client application `.jar` file, `com.sun.test.TestAppClient` is the full Java package name of the main client application, `scott` and `sample` are arguments to pass to the application, and `sun-acc.xml` is the name of the client configuration XML file. If `sun-acc.xml` is not in the current directory, you must give the absolute path location; otherwise the relative path is used. The relative path is relative to the directory where the command is being executed.

**ATTRIBUTES**

See attributes(5) for descriptions of the following attributes:
### appclient(1AS)

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

**SEE ALSO** package-appclient(1AS), asadmin(1AS)
asadmin(1AS)

NAME asadmin – utility for performing administrative tasks for the Sun ONE Application Server

SYNOPSIS

`asadmin subcommand [-short_option [short_option_argument]]*` 
`[-long_option [long_option_argument]]* [operand]*`

DESCRIPTION

Use the asadmin utility to perform any administrative task for the Sun ONE Application Server. You can use this utility in place of using the Administrator interface.

The `subcommand` identifies the operation or task you wish to perform. Subcommands are case-sensitive. Short option arguments have a single dash (`-`); while long option arguments have two dashes (`--`). Options modify how the utility performs a subcommand. Options are also case-sensitive. Most options require argument values except boolean options which toggle to switch a feature ON or OFF. Operands appear after the argument values, and are set off by a space, a tab, or double dashes (`--`). The asadmin utility treats anything that comes after the options and their values as an operand.

asadmin can be used in command shell invocation or multi command mode (known as multimode). In command shell invocation you invoke the asadmin utility from your command shell. asadmin executes the command, then exits. In multiple command mode, you invoke asadmin once, it then accepts multiple commands until you exit asadmin and return to the normal command shell invocation. Environment variables set while in multiple command mode are used for all subsequent commands until you exit multimode. You may provide commands by passing a previously prepared list of commands from a file or standard input (pipe). Additionally, you can invoke multimode from within a multimode session; once you exit the second multimode environment, you return to your original multimode environment.

You can also run the asadmin utility in interactive or non-interactive options. By default, the interactive option is enabled. It prompts you for the required arguments. You can use the interactive option in command shell invocation under all circumstances. You can use the interactive option in multimode when you run one subcommand at a time from the command prompt; and when you run in multimode from a file. Subcommands in multimode, when piped from an input stream, and subcommands invoked from another program, cannot run in the interactive option.

Local subcommands can be executed without the presence of an administration server. However, it is required that the user be logged into the machine hosting the domain in order to execute the subcommand and have access (permissions) for the installation and domain directories.

Remote subcommands are always executed by connecting to an administration server and executing the subcommand there. A running administration server is required. A user, however, can be on a local machine and execute a remote subcommand by connecting to a local administration server instance running on the machine. All remote subcommands require the `--host`, `--port`, `--user`, and `--password` options to be set, either on the command line or in the environment.
For subcommands that can be executed locally or remotely, if any one of the options
--host, --port, --user, or --password are set, either in the environment or in the
command line, the subcommand will run in remote mode.

Additionally, for subcommands that can be executed locally or remotely, if the
--local option is set to true, the subcommand will run locally. Also, if none of the
options --host, --port, --user, or --password are set, either on the command
line or in the environment, the subcommand is executed locally by default.

Setting the --local option to true overrides the --host, --port, --user, and
--password settings, even if specified. The subcommand will run in local mode.

Subcommands that can be executed locally accept the --domain option to specify the
domain of interest which assumes the domain as the default domain if there is only
one. If there is more than one domain, the --domain option is a required option.

For subcommands that can be run locally or remotely, when run remotely with the
--host, --port, --user, and --password options specified, the --domain option
is ignored. The --domain option is ignored if the subcommand will be run in remote
mode. Note that there is one administration instance per domain, so on a single
machine with multiple domains, local execution must specify the domain, and remote
execution must specify the --host, --port, --user, and --password options for
the administration instance for that domain.

For security purposes, you can set the password for a subcommand from a file instead
of entering the password at the command line. The --passwordfile option takes
the file containing the passwords. The valid contents for the file are:

AS_ADMIN_PASSWORD=value
AS_ADMIN_ADMINPASSWORD=value
AS_ADMIN_USERPASSWORD=value

Given the --passwordfile option and its value, the password options in the
passwordfile are exported to the global environment; subsequent subcommands
without the password options take this value. However, if both the --password and
--passwordfile options are specified on the command line, the password value in
the passwordfile is exported to the global environment and subsequent
subcommands without the --password option would take this value. However, for
the current subcommand, the --password option value specified on the command
line is taken since the --password option takes precedence over the
--passwordfile option.

To access the manpages for the Sun ONE Application Server Command-line interface
subcommands, add $AS_INSTALL/man to your MANPATH environment variable.

You can obtain overall usage information for any of the asadmin utility
subcommands by invoking the --help option. If you specify a subcommand, the
usage information for that subcommand is displayed. Using the help option without
a subcommand displays a listing of all the available subcommands.
See the *Sun ONE Application Server Administrator’s Guide* for a listing of all the options in their short form.

The environment variables are name/value pairs that can be set at any time and are in effect for the duration of the `asadmin` invocation. The `asadmin` utility will only read environment variables that have been exported using the `export` subcommand. Of course, environment subcommands are relevant only for the multiple subcommand mode (`multimode`).

### Environment Subcommands

- `export [name=value [name=value] *]`
  - Marks a variable name for automatic export to the environment of subsequent subcommands. All subsequent subcommands use the variable name values as specified; unless you `unset` them or `exit multimode`. If no arguments are specified, a list of all the exported variables and their values is displayed.
  - Exported shell environment variables set prior to invoking the `asadmin` utility are imported automatically and set as exported variables within `asadmin`. Unexported environment variables cannot be read by the `asadmin` utility.

- `unset env_var[env_var]*`
  - Removes one or more variables from the environment. The variables and their associated values no longer exist.
  - This subcommand can be run remotely only.

### Domain Administration Subcommands

  - Runs multiple commands without exiting the `asadmin` utility.
  - All variables are retained between subcommand invocations. Subcommand invocation is faster because `asadmin` does not need to start up each time.
  - Subcommands will be executed in `multimode` until the `exit` or `quit` command is given; at which point the `multimode` subcommand will exit.
  - You can provide subcommands by passing a previously prepared list of subcommands from a file or standard input (pipe).
  - You can invoke `multimode` from within a `multimode` session; once you exit the second `multimode` environment, you return to your original `multimode` environment.

The domain subcommands enable the configuration and management of a single administration server, and one or more associated J2EE server instances it controls. The domain encompasses all the data in the configuration repository for the administered instances, as well as all the deployed application data pertaining to the instances. Each administrative domain contains a unique administration server instance with its own unique set of port numbers.

A domain is constrained to a single machine; and domain names must be unique within the machine they are hosted on.

- `create-domain[--path domain_path] [--sysuser sys_user] [--passwordfile filename]
  --adminport port_number --adminuser admin --adminpassword password`
domain_name

- This subcommand can be run locally only.
- The sys_user must be a valid user on the system (Solaris only).
- The port_number cannot be currently active.
- The domain_name must be unique.
- The directory domain_path/domain_name must not already exist. The default domain will be created under $AS_DOMAINS_PATH directory.

delete-domain domain_name

- This subcommand can be run locally only.
- The domain must already exist, but the instances within the domain must not be executing.

start-domain [--domain domain_name]

- This subcommand can be run locally only.
- The domain must currently exist on the local machine.


- This subcommand can be run both locally and remotely. The domain must exist on the local machine to run this subcommand locally.

list-domains [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--passwordfile filename] [--secure|-s]

- This subcommand can be run both locally and remotely.
- Set the option --local to true to execute this subcommand locally. If running remotely, the administrative server must be running on the hostname specified.
- One or more domain must already exist.

These subcommands configure the instances that the clients may control or manage.


- The named instance must not exist within that domain.


- This subcommand can be run both locally and remotely.
- To start locally, with a domain name identified, the named instance must already exist within that domain.
To start remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server.


This subcommand can be run both locally and remotely.

The server instance must not be running before you can delete it.

To delete remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server.

Use with discretion since this subcommand is destructive and there is no undo.


This subcommand can be run both locally and remotely.

The named instance must already exist within the given domain; and the instance must be running.


This subcommand is not supported on Windows.

This subcommand can be run both locally and remotely.

To restart remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server, and the instance must be running.

list-instances [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename] [--secure|-s]

This subcommand can be run both locally and remotely.

To list remote instances, the named administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server.

start-appserv

This subcommand can be run locally only.

One or more domain must already exist.

Starts all the domains defined for the application server installation; use with caution.
stop-appserv

- Stops all the domains, and its instances, in the application server installation; use with caution.
- This subcommand can be run locally only.
- One or more domain must already exist.

These subcommands display the list of instances/services in the server, the status of the instance, and the service of a deployed application on the server.

show-instance-status --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--passwordfile filename] [--secure|-s] instance_name

- The instance must already exist. If the instance does not exist, the subcommand fails.
- The status is a string representation returned by the server; it can be starting/started, or stopping/stopped.

show-component-status --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] component_name

- Gets the status of the deployed component.

list-components --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type application|ejb|web|connector] instance_name

- Lists all components for the specified instance.
- If the --type is not specified, then all the deployed applications and standalone modules are listed.

list-sub-components --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type ejbs|servlets] [--instance instance_name] [--appname app_name] module_name

- Lists your EJBs or Servlets in a deployed module or in a module of the deployed application.
- If the module is not identified, all modules are listed.
- The component type defaults to EJBs.

enable --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type application|ejb|web|connector] [--instance instance_name] component_name

- If the component is already enabled, then it is re-enabled.
- The component must have been deployed in order to be enabled. If it has not been deployed, an error message is returned.
- --type identifies the type of deployed component.

disable --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type application|ejb|web|connector]
### Deployment Subcommands

- `--instance instance_name` `component_name`
  - Immediately stops the named component.
  - The component must have been deployed to the specified instance. If the component has not been deployed, an error message is returned.
  - `--type` identifies the type of deployed component.

These subcommands are used for deploying applications and modules to the named instance on the Sun ONE Application Server.

```bash
```

- Deploys the named component of the specified type. If the component does not exist, the system indicates accordingly. If the component is already deployed or already exists, it is forcefully re-deployed if the `force` option is set to true.
- `--contextroot` is valid only if the archive is a web-module.
- `--name` is the name of the deployable component.
- If `upload` is set to true, the system uploads the deployable file to the administration server.
- The deployable file location should be an absolute path on the server machine when the `upload` option is set to true.

```bash
```

- Deploys the J2EE component that is in the directory located on the server machine.
- `--force` option makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.
- `--contextroot` is valid only if the archive is a web-module. Ignored for other archive types; defaults to `filename_without_extension`.

### Configuration Subcommands

  - Removes the component from the named instance.

These subcommands allow you to access the attributes of the configurable entities in the Sun ONE Application Server.

```bash
```
When using the wildcard character to get multiple attribute values while in single mode, enclose the attribute in double quotes. In multimode, DO NOT use the double quotes.

--monitor defaults to false. If set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.

See the Sun ONE Application Server Administrator’s Guide for a listing of the valid attribute names.

```
set [--monitor] --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [-s attribute_name=value [attribute_name=value]...]
```

Sets the values of one or more configurable attribute. The settings do not take affect until you run the reconfig subcommand.

```
```

Applies the changes you have made for a server instance.

--discardmanualchanges defaults to false. When set to true, discards the changes made manually to the server.xml file.

--keepmanualchanges defaults to false. When set to true, allows the manual changes made to the server.xml file to take affect.

--discardmanualchanges=false is NOT equal to --keepmanualchanges=true. --discardmanualchanges=false is actually equal to not specifying the option. An error message is displayed if both options are set to false or not specified and a manual change has been made to the server.xml file.

Use this subcommand with discretion since there is no undo, and the changes applied are made directly to your server.xml file.

```
list [--monitor] --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [-s] element_name
```

Lists the configurable or monitorable elements (child nodes).

--monitor defaults to false. If set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.

These subcommands are used to administer the IMQ server of the Sun ONE Application Server.

```
create-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [-s] [-instance instance_name] --desttype type [--property (name=value) [name=value]...] dest_name
```

Valid values for the destination type include: topic and queue.

Valid values for destination name is the name of the JMS destination. Valid value is any name that can be a Java identifier.

The name/value property pairs are used to name JMS specific attributes to further customize the destination being created.
delete-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] --desttype type dest_name

- Valid values for the destination type include: topic and queue.
- Valid values for destination name is the name of the JMS destination. Valid value is any name that can be a Java identifier.
- Destroys the named destination.

list-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--desttype type] instance_name

- Valid values for the destination type include: topic and queue.
- Lists the named JMS destinations.

jms-ping --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name

- Checks to see if the JMS provider is up and running for the named instance.

The Resource Administration subcommands allow you to manage the various resources.


- --datasourceclassname is the name of the vendor supplied JDBC datasoure resource manager.
- --restype must be specified to disambiguate when a datasource class implements both interfaces. An error is produced when this option has a legal value and the indicated interface is not implemented by the datasource class. This option does not have a default value.
- --steadypoolsize is the minimum and initial number of connections maintained in the pool.
- --maxpoolsize is the maximum number of connections that can be created.
- --maxwait is the amount of time a caller will wait before a connection timeout is sent. The default is 60 seconds. A value of 0 forces the caller to wait indefinitely.
- --poolresize is the number of connections to be removed when idletimeout timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches steadypoolsize, the connection removal stops.
- --idletimeout is the maximum time (in seconds) that a connection can remain idle in the pool. After this time, the implementation can close this connection. It is recommended that this timeout is kept shorter than the server side timeout to
prevent the accumulation of unusable connections in the application.

- **-isolationlevel** specifies the transaction-isolation-level on the pooled database connections. This option does not have a default value. If not specified, the pool operates with default isolation level provided by the JDBC driver. A desired isolation level can be set using one of the standard transaction isolation levels: read-uncommitted, read-committed, repeatable-read, serializable. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool. This could lead to program errors.

- **-isisolationguaranteed** is applicable only when a particular isolation level is specified for transaction-isolation-level. The default value is true. This assures that every time a connection is obtained from the pool, it is guaranteed to have the isolation set to the desired value. This could have some performance impact on some JDBC drivers. Set this option to false if you are certain that the application does not change the isolation level before returning the connection.

- **-isconnectvalidatereq** if set to true connections are validated (checked to see if they are usable) before giving out the application. The default is false.

- **-validationmethod** is the name of the validation table used to perform a query to validate a connection.

- **-validationtable** is the name of the validation table used to perform a query to validate a connection. This parameter is mandatory if connection-validation-type is set to table. Verification by accessing a user specified table may become necessary for connection validation.

- **-failconnection** if set to true, all connection in the pool must be closed if a single validation check fails; defaults to false. One attempt is made to re-establish failed connections.

- **-description** is the text description of the JDBC connection pool.

- **-property** is the optional attribute/value pairs for configuring the connection pool.

```bash
```

- Removes the JDBC connection pool from the named instance.

```bash
list-jdbc-connection-pools --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

- Gets all the JDBC connections pools for the named instance.

```bash
create-jdbc-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] --instance instance_name
```

- **-connectionpoolid** is the name of the JDBC connection pool. If two or more JDBC resource elements point to the same connection pool element, the same pool connections are used at runtime.

- **-enabled** determines if the resource is enabled at runtime.
--description is the text description of the JDBC connection pool.

delete-jdbc-resource --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-instance instance_name]jndi_name

- Removes the JDBC resource with the given JNDI name from the specified instance.

list-jdbc-resources --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s]instance_name

- Gets all the JDBC resources from the specified instance.

create-jms-resource --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-instance instance_name][-resourcetype type][-enabled=true][-description text][-property (name=value)[|name=value]*]jndi_name

- --resourcetype is the JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.
- --enabled determines if the resource is enabled at runtime.
- --property is the optional attribute/value pairs for configuring the JMS resource.

delete-jms-resource --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-instance instance_name]jndi_name

- Removes the JMS resource with the given JNDI name from the specified instance.

list-jms-resources --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-resourcetype type]instance_name

- Gets all the JMS resources for the named resource type from the specified instance.

create-jndi-resource --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-instance instance_name][-jndilookupname lookup][-resourcetype type][-factoryclass class_name][-enabled=true][-description text][-property (name=value)[|name=value]*]jndi_name

- --jndilookupname is the lookup name used by the external container.
- --resourcetype is the JNDI resource type which can be: topic or queue.
- --factoryclass is the class that creates the JNDI resource.
- --enabled determines if the resource is enabled at runtime.
- --property is the optional attribute/value pairs for configuring the JNDI resource.

delete-jndi-resource --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-instance instance_name]jndi_name

- Removes the JNDI resource with the given JNDI name from the specified instance.
list-jndi-resources --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name

- Gets all the JNDI resources from the specified instance.

create-javamail-resource --user admin_user[--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
[--instance instance_name] --mailhost hostname --mailuser username
--fromaddress address [--storeprotocol imap] [--storeprotocolclass com.sun.mail.imap.IMAPStore]
[--transprotocol=smtp] [--transprotocolclass=com.sun.mail.smtp.SMTPTransport]
[--debug=false] [--enabled=true] [--description text]
[--property (name=value):name=value]* jndi_name

- --debug if set to true, the server starts up in debug mode for this resource.
- --enabled determines if the resource is enabled at runtime.
- --property is the optional attribute/value pairs for configuring the JNDI resource.

delete-javamail-resource --user admin_user[--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
[--instance instance_name] --mailhost hostname --mailuser username
--fromaddress address --storeprotocol imap --storeprotocolclass com.sun.mail.imap.IMAPStore
[--transprotocol=smtp] [--transprotocolclass=com.sun.mail.smtp.SMTPTransport] [--debug=false]
[--enabled=true] [--description text] [--property (name=value):name=value]* jndi_name

- --debug if set to true, the server starts up in debug mode for this resource.
- --enabled determines if the resource is enabled at runtime.
- --property is the optional attribute/value pairs for configuring the JNDI resource.

list-javamail-resources --user admin_user[--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] instance_name

- Gets all the Javamail resources from the specified instance.

create-persistence-resource --user admin_user[--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
[--instance instance_name] --jdbcjndiname jndi_name --factoryclass classname
[--enabled=true] [--description text]
[--property (name=value):name=value]* jndi_name

- --jdbcjndiname is the JDBC resource used to obtain the database connections.
  This must be the name of one of the pre-created JDBC resources.
- --enabled determines if the resource is enabled at runtime.
- --property is the optional attribute/value pairs for configuring the resource.

delete-persistence-resource --user admin_user[--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
[--instance instance_name] jndi_name

- Removes the persistence resource with the given JNDI name from the specified instance.

list-persistence-resources --user admin_user[--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
I Gets all the persistence resources from the specified instance.
create-custom-resource --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] --resourcetype type --factoryclass classname [--enabled=true] [--description text] [--property (name=value) [name=value]*] jndi_name

--enabled determines if the resource is enabled at runtime.
--property is the optional attribute/value pairs for configuring the resource.
delete-custom-resource --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [-s] [--instance instance_name] jndi_name

I Removes the custom resource with the given JNDI name from the specified instance.
list-custom-resources --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name]

I Gets all the custom resources from the specified instance.
add-resources --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name] xml_file_path

I Registers the named resource in the XML file identified.
I The xml_file_path is the path to the XML file containing the resources to be registered.

IIOP Listeners Subcommands

The IIOP Listeners subcommands allow you to manage the listener resources.
create-iiop-listener --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name] --listeneraddress address[-iiopport iiop_port] [--enabled=true] [--property (name=value) [name=value]*] listener_ID

--enabled determines if the resource is enabled at runtime.
--property is the optional attribute/value pairs for configuring the resource.
delete-iiop-listener --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name] listener_ID

I Removes the custom resource with the given IIOP listener from the specified instance.
list-iiop-listeners --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name]

I Gets all the IIOP listeners from the specified instance.

Lifecycle Module Subcommands

Lifecycle module subcommands enable you to run short or long duration Java-based tasks within the Application Server environment.
create-lifecycle-module --user admin_user[--password admin_password] [--host localhost] [-s] [--instance instance_name]
--classname class_name [--classpath classpath] [--loadorder load_order] [--failurefatal failure_fatal] [--enabled=true] [--property (name=value) [name=value]]* module_name

- `--loadorder` is an integer value used to force the order in which deployed lifecycle modules are loaded at server startup. Smaller numbered modules get loaded sooner. The order is unspecified if two or more lifecycle modules have the same load-order value.

- `--failurefatal` if true indicates abort server startup if the module does not load properly.

- `--enabled` determines if the resource is enabled at runtime.

- `--property` is the optional attribute/value pairs for configuring the resource.

```
display-lifecycle-module --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] module_name
```

- Removes the lifecycle module with the given module name from the specified instance.

```
display-lifecycle-modules --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

- Gets all the lifecycle modules from the specified instance.

The server determines the MIME type of a requested resource by invoking the type-by-extension directive.

```
```

- `--mimefile` is the name of a MIME types file.

- `mime_ID` is the internal name for the MIME types listing. It is used in a virtual server element to define the MIME types used by the virtual server.

```
```

- Removes the MIME with the given MIME ID from the specified instance.

```
list-mimes --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

- Gets all the MIMEs from the specified instance.

The HTTP listener subcommands allow you to connect between the server and clients.

```
```

HTTP Listener Subcommands

MIME Subcommands

asadmin(1AS)

Application Server Utility  31
--listenerport is the port number to create the listen socket on. Legal values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges. Configuring an SSL listen socket to listen on port 443 is recommended.

--defaultvsv is the ID attribute of the default virtual server for this particular connection group.

--servername identifies to the server what to put in the hostname section of any URLs sent to the client. This affects URLs the server automatically generates; it does not affect the URLs for directories and files stored in the server. If your server uses an alias, this name should be the alias name. If a colon and port number is appended, that port is used in URLs that the server sends to the client.

--family is the socket family type; defaults to inet. Legal values are: inet, inet6, and nca. Use the value inet6 for IPv6 listen sockets. When using the value of inet6, IPv4 addresses are prefixed with ::ffff: in the log file. Specify nca to make use of the Solaris Network Cache and Accelerator.

--acceptorthreads is the number of acceptor threads for the listen socket. The recommended value is the number of processors in the machine.

--blockingenabled determines whether the HTTP listener socket and the accepted socket are put into blocking mode. Use of blocking mode may improve benchmark scores.

--securityenabled determines whether the HTTP listener runs SSL. You can turn SSL2 or SSL3 ON or OFF and set ciphers using an SSL element. The security setting in the init.conf file globally enables or disables SSL by making certificates available to the server instance. Therefore, security in the init.conf file must be ON or security in the server.xml file does not work.

--enabled determines if the resource is enabled at runtime.


Removes the HTTP listener with the given HTTP listener ID from the specified instance.


Gets all the HTTP listeners from the specified instance.

The HTTP quality of service subcommands allow you to define the quality of service parameters on the HTTP path.

- **--virtualserver** is the virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

- **--bwlimit** is the maximum bandwidth limit, for the virtual server class or virtual server, in bytes per second. The default is no limit.

- **--enforcebwlimit** determines whether the bandwidth limit should be enforced or not.

- **--connlimit** is the maximum number of concurrent connections for the server, virtual server class, or virtual server.

- **--enforceconnlimit** determines whether the connection limit should be enforced or not.

```plaintext
```

Removes the HTTP QOS with the given virtual server ID from the specified instance.

The authorization database subcommands define the user database used by the virtual server.

```plaintext
```

- **--database** is the user database name in the dbswitch.conf file.

- **--virtualserver** is the virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

- **--basedn** overrides the base DN lookup in the dbswitch.conf file. However, the basedn value is still relative to the base DN value from the dbswitch.conf entry.

- **--certmaps** is the certificate to LDAP entry mappings as defined in the certm.conf file. If not present, all mappings are used. All lookups are based on mappings in the certmap.conf file and are relative to the final base distinguished name (DN) of the virtual server.

- **authdb_ID** is the user database name in the virtual server’s ACL file.

```plaintext
```

- **--virtualserver** is the virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

- **authdb_ID** is the user database name in the virtual server’s ACL file.

```plaintext
```

These subcommands allow for the management and manipulation of user databases within an application server environment, providing detailed control over bandwidth limits, connection limits, and database configurations.
---virtualserver is the virtual server ID. It can also be referred to as the variable \$id in an obj.conf file. A virtual server ID cannot begin with a number.

**authdb_ID** is the user database name in the virtual server's ACL file.

The authorization realm subcommands define the user realm used by the virtual server.

```
create-auth-realm --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--instance instance_name] --classname realm_class [--property (name=value) ] auth_realm_name
```

- **--classname** is the Java class which implements this realm.
- **--property** name/value pairs of provider implementation specific attributes.

```
delete-auth-realm --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--instance instance_name] auth_realm_name
```

- Removes the authorization realm with the given authorization name from the specified instance.

```
list-auth-realms --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--instance instance_name]
```

- Gets all the authorization realms from the specified instance.

The access control list subcommands allow you to manage and define the ACL file used by the virtual server.

```
```

- The ACL_ID is the internal name for the ACL file listing. This ID is used in a virtual server element to define the ACL file used by the virtual server.

```
```

- Removes the ACL with the given ACL ID from the specified instance.

```
list-acls --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--instance instance_name]
```

- Gets all the ACLs from the specified instance.

Virtualization in the Application Server allows multiple URL domains to be served by the same HTTP server process which is listening on multiple host addresses. If the application is available at two virtual servers, they still share the same physical resource pools.

```
```

Virtualization in the Application Server allows multiple URL domains to be served by the same HTTP server process which is listening on multiple host addresses. If the application is available at two virtual servers, they still share the same physical resource pools.
asadmin(1AS)

[--state on] [--acls acls] [--acceptlang=false] [--logfile log_file]
[--property (name=value) [name=value]*] virtual_server_ID

- **--hosts** is a comma-separated list of values allowed in the host request header to select the current virtual server. Each virtual that is configured to the same connection group must have a unique hosts value for that group.
- **--mime** is the ID of the mime element used by the virtual server.
- **--httplisteners** is a comma-separated list of HTTP listener IDs. Required only for a virtual server that is not the default virtual server.
- **--defaultwebmodule** is the standalone web module associated with the named virtual server.
- Use the **--configfile** option to change the default virtual server initialization from $AS_instance_root/config/obj.conf to the named configuration file.
- **--defaultobj** names the object loaded from an obj.conf file which is default. The default object is expected to have all the name translation directives for the virtual server. Any server behavior that is configured in the default object affects the entire virtual server class.
- **--state** determines whether a virtual server is active (on) or inactive (off or disabled). Default is active (on). When inactive, the virtual server does not service requests.
- **--acls** is a comma-separated list of ID attributes of ACL elements. Specifies the ACL files used by the virtual server.
- **--acceptlang** when turned on, the server parses the Accept-Language header and sends an appropriate language version based on which language the client can accept. Set this value to ON only if the server supports multiple languages. The default setting is determined from the virtual-server-class.
- **--logfile** name of the file where the log has to be written to.

delete-virtual-server --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [-s] [instance_name] virtual_server_ID

- Deletes a virtual server with the given virtual server ID.

list-virtual-servers --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [-s] instance_name

- Lists all the virtual servers in the named instance.

create-profiler --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [-s] [instance_name] [classpath classpath|--nativelibpath native_library_path] [enabled=true]
[--property (name=value) [name=value]*] profiler_name

- **--classpath** is the Java classpath string that specifies the classes needed by the profiler.
- **--nativelibpath** is automatically constructed to be a concatenation of the Application Server installation relative path for its native shared libraries, standard JRE native library path, the shell environment setting $LD_LIBRARY_PATH on
UNIX) and any path that may be specified in the profile element.

- --property name/value pairs of provider specific attributes.

```
delete-profiler --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

- Deletes a profiler for the given instance.

```
list-profilers --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

- Gets all the profilers in the given instance.

### SSL Subcommands

The SSL subcommand allow you to manage the SSL elements in the HTTP listener or IIOP listener.

```
```

- --type is the type of service or listener that the SSL is created for. The type can be: http-listener, iiop-listener, and iiop-service.

- --certname is 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 in this attribute is optional.

- --ssl2enabled determines whether SSL2 is enabled.

- --ssl2ciphers is a comma separated list of the SSL2 ciphers used. Use the prefix + to enable or — to disable. Allowed values are: rc4, rc4export, rc2, rc2export, idea, des, desede3. If no value is specified, all supported ciphers are assumed to be enabled.

- --ssl3enabled determines whether SSL3 is enabled.

- --ssl3ciphers is a comma separated list of the SSL3 ciphers used. Use the prefix + to enable or — to disable. Allowed values are: rsa_rc4_128_md5, rsa3des_sha, rsa_des_sha, rsa_rc4_40_md5, rsa_rc2_40_md5, rsa_null_md5. Allowed TSL values are: rsa_des_56_sha, rsa_rc4_56_sha. If no value is specified, all supported ciphers are assumed to be enabled.

- --tlsenabled determines whether TLS is enabled.

- --tlsrollbackenabled determines whether TLS rollback is enabled. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5.

- --clientauthenabled determines whether SSL3 client authentication is performed on every request independent of ACL-based access control.

```
delete-ssl --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] --type [http-listener|iiop-listener|iiop-service] [--instance instance_name] [listener_id]
```

- Removes a SSL from the given instance.
--type is the type of service or listener that the SSL is created for. The type can be: http-listener, iiop-listener, and iiop-service.

The JVM Options subcommands allow you to manage the options in the Java configuration or profiler elements of the server.xml file.

```
create-jvm-options --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] [--profiler=false] [jvm_option_name=jvm_option_value] [:jvm_option_name=jvm_option_value]*
```

- jvm_option_name=jvm_option_value is the JVM option name and the JVM option value associated with it. You can enter more than one JVM option separated by a colon (:). If the JVM option starts with a dash (-) then use two dashes (—) before the operand to distinguish that JVM option is an operand and not an option.

- --profiler indicates if the JVM options are for the profiler. The profiler must exist for this option to be true.

- JVM options are used to record the settings needed to get a particular profiler going.

```
delete-jvm-options --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--instance instance_name] [--profiler=false] [jvm_option_name=jvm_option_value] [:jvm_option_name=jvm_option_value]*
```

- Deletes the JVM options from the Java configuration or profiler elements.
- You can enter more than one JVM option separated by a colon (:). If the JVM option starts with a dash (-) then use two dashes (—) before the operand to distinguish that JVM option is an operand and not an option.

```
install-license
```

- This subcommand can be run locally only.
- Displays the license agreement allowing you to accept/reject the license terms.

```
display-license [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--instance instance_name] [--userpassword user_password] [--groups user_groups:*] user_name
```

- This subcommand can be run both locally and remotely.
- Displays the license terms currently in effect.

```
create-file-user --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] [--userpassword user_password] [--groups user_groups:*] user_name
```

- --userpassword is the password for the file user.
- --groups is the group that the file user belongs to.
- user_name is the name of the file user to be created.
- You can enter more than one user group separated by a colon (:).

```
delete-file-user --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] user_name
```

- Deletes the named file user associated with the specified instance.

```
update-file-user --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
```

- --userpassword is the password for the file user.
- --groups is the group that the file user belongs to.
- user_name is the name of the file user to be updated.
- You can enter more than one user group separated by a colon (:).
asadmin(1AS)

[-userpassword user_password] [-groups user_groups[:user_groups]*] user_name

- -userpassword is the password for the file user.
- -groups is the group that the file user belongs to.
- user_name is the name of the file user to be updated.
- You can enter more than one user group separated by a colon (:). If the user group starts with a dash (-) then use two dashes (--) before the operand to distinguish that group option is an operand and not an option.

list-file-users --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s]instance_name
- List all the file users associated with the named instance.

list-file-groups --user admin_user[-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s][-name user_name]instance_name
- List all the file groups associated with the named instance.
- -name is the name of the file user.

shutdown[--user admin_user][-password admin_password][-host localhost] [-port 4848][-passwordfile filename][-secure|-s]
- Gracefully shuts down the administration server, all its domains, and all the running instances.

version[--user admin_user][-password admin_password][-host localhost] [-port 4848][-local=false][-verbose=false][-passwordfile filename][-secure|-s]
- displays the version information for the Sun ONE Application Server and the Command-line interface.

help [subcommand]
- displays the syntax for the named subcommand. If the subcommand is not specified displays the syntax of all the Command-line interface subcommands.

ATTRIBUTES
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Unstable</td>
</tr>
</tbody>
</table>

SEE ALSO
appclient(1AS), package-appclient(1AS)
asadmin-add-resources

### NAME
asadmin-add-resources, add-resources – registers the named resource in the XML file specified.

### SYNOPSIS
```
add-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] xml_file_path
```

### DESCRIPTION
Registers the named resource in the XML file specified. The `xml_file_path` is the path to the XML file containing the resources to be registered.

### OPTIONS
- **--user**
  administrative user associated for the instance.
- **--password**
  administrative password corresponding to the administrative user.
- **--host**
  host name of the machine hosting the administrative instance.
- **--port**
  administrative port number associated with the administrative host.
- **--secure**
  indicates communication with the administrative instance in secured mode.
- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).
- **--instance**
  name of the instance.

### OPERANDS
- **xml_file_path**
  path to the XML file containing the resource(s) to be registered.

### EXAMPLES
**EXAMPLE 1 Using add-resources**
```
asadmin> add-resources --user admin --passwordfile passwords.txt --host localhost --port 4848 --instance server1 resource.xml
```
Created the resource

Where: `resource.xml` is the resource created.

### EXIT STATUS
- **0**
  command executed successfully
- **1**
  error in executing the command

### INTERFACE
Resources folder

### EQUIVALENT
```
asadmin-create-jdbc-connection-pool(1AS),
asadmin-list-jdbc-resource(1AS), asadmin-create-jms-resource(1AS),
asadmin-create-jndi-resource(1AS),
asadmin-create-javamail-resource(1AS),
asadmin-create-persistence-resource(1AS),
asadmin-create-custom-resource(1AS)
```

### SEE ALSO
Application Server Utility 39
asadmin-create-acl(1AS)

NAME
asadmin-create-acl, create-acl – adds a new access control list file for the named instance

SYNOPSIS

DESCRIPTION
Gets the access control lists associated with the named server instance.

OPTIONS
- --user administrative user associated for the instance.
- --password administrative password corresponding to the administrative user.
- --host host name of the machine hosting the administrative instance.
- --port administrative port number associated with the administrative host.
- --secure indicates communication with the administrative instance in secured mode.
- --passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
- --instance name of the instance.
- --aclfile name of the default acl file.

OPERANDS
acl_ID internal name for the ACL file listing. This ID is used in a virtual server element to define the ACL file used by the virtual server.

EXAMPLES
EXAMPLE 1 Using create-acl
asadmin> create-acl --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --aclfile "/export/sample_acl_file.scl" sampleACL
Created ACL with id=sampleACL

Where: sampleACL is the name of the ACL created.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
Access Control List page

EQUIVALENT
SEE ALSO
asadmin-delete-acl(1AS), asadmin-list-acls(1AS)

40 Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
asadmin-create-authdb(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-create-authdb, create-authdb – adds the new authorized database for the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>Adds the named authorized database associated with the named server instance.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td><strong>--user</strong> administrative user associated for the instance.</td>
</tr>
<tr>
<td></td>
<td><strong>--password</strong> host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td><strong>--host</strong> host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td><strong>--port</strong> administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td></td>
<td><strong>--passwordfile</strong> file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td></td>
<td><strong>--secure</strong> if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td></td>
<td><strong>--instance</strong> name of the instance.</td>
</tr>
<tr>
<td></td>
<td><strong>--database</strong> user database name in the dbswitch.conf file.</td>
</tr>
<tr>
<td></td>
<td><strong>--virtualserver</strong> virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.</td>
</tr>
<tr>
<td></td>
<td><strong>--basedn</strong> overrides the base DN lookup in the dbswitch.conf file. However, the basedn value is still relative to the base DN value from the dbswitch.conf entry.</td>
</tr>
<tr>
<td></td>
<td><strong>--certmaps</strong> certificate to LDAP entry mappings as defined in the certm.conf file. If not present, all mappings are used. All lookups are based on mappings in the certmap.conf file and are relative to the final base distinguished name (DN) of the virtual server.</td>
</tr>
<tr>
<td>OPERANDS</td>
<td><strong>authdb_id</strong> user database name in the virtual server’s ACL file.</td>
</tr>
</tbody>
</table>

**EXAMPLE 1** Using create-authdb

    asadmin> create-authdb --user admin --password adminadmin --host fuyako --port 7070 --database default --virtualserver server1 --basedn "o=sun" sampleAuth
    Created AuthDB with id = sampleAuth

Where sampleAuth is the authdb created.
### asadmin-create-authdb(1AS)

<table>
<thead>
<tr>
<th>EXIT STATUS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

### INTERFACE EQUIVALENT

unknown

### SEE ALSO

asadmin-delete-authdb(1AS), asadmin-list-authdbs(1AS)
NAME
asadmin-create-auth-realm, create-auth-realm – adds the new authorized realm for the named instance

SYNOPSIS
create-auth-realm --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename] [--secure|-s] [--instance instance_name] --classname realm_class[--property (name=value)][:name=value]*auth_realm_name

DESCRIPTION
Adds the named authorized realm associated with the named server instance.

OPTIONS
--user行政 user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--classname Java class which implements this realm.
--property optional attributes name/value pairs of provider implementation specific attributes.

OPERANDS
auth_realm_name name of this realm.

EXAMPLES
EXAMPLE 1 Using create-auth-realm
asadmin> create-auth-realm --user admin --password adminadmin
--host bluestar --port 4848 --instance server1 --classname com.iplanet.ias.security.auth.realm.DB.Database --property defaultuser=admin:Password=admin db
Created Auth realm with id = db

Where db is the auth realm created.

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-delete-auth-realm(1AS), asadmin-list-auth-realms(1AS)
### NAME
asadmin-create-custom-resource, create-custom-resource – registers the custom resource to the named instance

### SYNOPSIS
```
create-custom-resource --user admin_user --password admin_password
                      [--host localhost] [--port 4848] [--passwordfile filename]
                      [--secure|-s] [--instance instance_name] --resourcetype type
                      --factoryclass classname [--enabled=true] [--description text]
                      [--property (name=value) [value=value] *] jndi_name
```

### DESCRIPTION
Registers the custom resource to the named instance.

### OPTIONS
- **--user**
  administrative user associated for the instance.
- **--password**
  administrative password corresponding to the administrative user.
- **--host**
  host name of the machine hosting the administrative instance.
- **--port**
  administrative port number associated with the administrative host.
- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).
- **--secure**
  if true, uses SSL/TLS to communicate with the administrative instance.
- **--instance**
  name of the instance.
- **--resourcetype**
  type of custom resource to be created.
- **--factoryclass**
  class that creates the custom resource.
- **--enabled**
  determines whether the resource is enabled at runtime or not.
- **--description**
  text description of the JDBC connection pool.
- **--property**
  optional attributes name/value pairs for configuring the resource.

### OPERANDS
- **jndi_name**
  JNDI name of the custom resource to be created.

### EXAMPLES
**EXAMPLE 1** Using create-custom-resources
```
asadmin> create-custom-resource --user admin --password adminadmin
 --host fuyako --port 7070 --instance server1 --resourcetype customType
 factoryclass "com.custom.class" --description "this is a sample of creating a custom resource" sample_custom_resource
Created the custom resource with jndiname = sample_custom_resource

Where sample_custom_resource is the custom resource created.
```

### EXIT STATUS
- **0**
  command executed successfully
error in executing the command

asadmin-create-custom-resource(1AS)

asadmin-delete-custom-resource(1AS),
asadmin-list-custom-resources(1AS)
asadmin-create-domain(1AS)

NAME

asadmin-create-domain, create-domain – creates a domain with the given name

SYNOPSIS

create-domain [--path domain_path] [--sysuser sys_user]
    [--passwordfile filename] --adminport port_number --adminuser admin
    --adminpassword password domain_name

DESCRIPTION

Use the create-domain command to create a domain with the specified administration port number, administration user, administration password, and domain name. By creating a domain, an administration server is created in a directory named as the domain name.

This command can be run locally only.

OPTIONS

--path directory path where the domain should be created. If specified, path must be accessible in the filesystem. If not specified, the domain is created under $AS_DOMAINS_PATH directory.

--sysuser owner of the domain directory. Must be a valid user on the system (Solaris only). The domain is created under the specified system username. If not specified, the current username is used.

--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).

--adminport port of the administrative instance. The port number cannot be currently active.

--adminuser user name associated with the administrative instance.

--adminpassword password associated with the administrative name.

OPERANDS

domain_name name of the domain. Must be a unique name.

EXAMPLES

EXAMPLE 1 Using create-domain

asadmin> create-domain --path /u/domain1/domain_root --sysuser user1 --adminuser admin --adminpassword adminpassword domain1

created domain domain1 successfully

Where: the domain1 domain is created in the domain_root directory.

EXIT STATUS

0 command executed successfully

1 error in executing the command

SEE ALSO

asadmin-delete-domain(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS), asadmin-list-domains(1AS), asadmin-multimode(1AS)

46 Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
asadmin-create-file-user(1AS)

NAME
asadmin-create-file-user, create-file-user - creates a new file user

SYNOPSIS
create-file-user --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] [--userpassword user_password] [--groups user_groups] [user_name]

DESCRIPTION
creates a new file user to the named user name.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--userpassword password for the file user.
--groups group where the file user belongs to.

OPERANDS
user_name name of file user to be created.

EXAMPLES
EXAMPLE 1 Using the create-file-user command to create a file user
asadmin> create-file-user --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 --userpassword sample
--groups staff:manager sample_user
Created File user sample_user

Where: the sample_user is the file user created.

EXAMPLE 2 Using the create-file-user command with the passwordfile option
asadmin> create-file-user --user admin --host fuyako --port 7070
--passwordfile sample_passwordfile --instance server1 --groups staff:manager
sample_file_user
Created File user sample_file_user

Where: sample_password file contains the following:

AS_ADMIN_PASSWORD=adminadmin
AS_ADMIN_USERPASSWORD=sample
EXAMPLE 2 Using the create-file-user command with the passwordfile option

AS_ADMIN_PASSWORD is the administrative password. AS_ADMIN_USERPASSWORD is the file user password.

EXIT STATUS

- 0 command executed successfully
- 1 error in executing the command

SEE ALSO

asadmin-delete-file-user(1AS), asadmin-list-file-users(1AS),
asadmin-update-file-user(1AS), asadmin-list-file-groups(1AS)
asadmin-create-http-listener

## NAME
asadmin-create-http-listener, create-http-listener – adds a new HTTP listener socket

## SYNOPSIS
```bash
```

## DESCRIPTION
Creates the HTTP listener associated with the named identifier.

## OPTIONS
- **--user**
  administrative user associated for the instance.
- **--password**
  administrative password corresponding to the administrative user.
- **--host**
  host name of the machine hosting the administrative instance.
- **--port**
  administrative port number associated with the administrative host.
- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).
- **--secure**
  if true, uses SSL/TLS to communicate with the administrative instance.
- **--instance**
  name of the instance.
- **--listenerport**
  port number to create the listen socket on. Legal values are 1–65535. On UNIX, creating sockets that listen on ports 1–1024 requires superuser privileges. Configuring an SSL listen socket to listen on port 443 is recommended.
- **--defaultvs**
  ID attribute of the default virtual server for this particular connection group.
- **--servername**
  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. This name should be the alias name if your server uses an alias. If a colon and port number is appended, that port will be used in URLs that the server sends to the client.
- **--family**
  socket family type; defaults to inet. Legal values are: inet, inet6, and nca. Use the value inet6 for IPv6 listen sockets. When using the value of inet6, IPv4
addresses are prefixed with ::ffff: in the log file. Specify nca to make use of the Solaris Network Cache and Accelerator.

```
--acceptorthreads
```

determines the number of acceptor threads for the listen socket. The recommended value is the number of processors in the machine.

```
--blockingenabled
```

determines whether the HTTP listener socket and the accepted socket are put into blocking mode. Use of blocking mode may improve benchmark scores.

```
--securityenabled
```

determines whether the HTTP listener runs SSL. You can turn SSL2 or SSL3 ON or OFF and set ciphers using an SSL element. The security setting in the `init.conf` file globally enables or disables SSL by making certificates available to the server instance. Therefore, security in the `init.conf` file must be ON or security in the `server.xml` file does not work.

```
--enabled
```

determines if the resource is enabled at runtime.

**OPERANDS**

`listener_id` listener ID of the HTTP listener.

**EXAMPLES**

**EXAMPLE 1 Using create-http-listener**

```
asadmin> create-http-listener --user admin --password adminadmin
--host fuyako --port 7070 --address 0.0.0.0 --instance server1 --listenerport 7272
--defaultvs server1 --servername fuyako.red.iplanet.com --family inet6
--acceptorthreads 2 --blockingenabled=true --securityenabled=false
--enabled=false sampleListener
Created HTTP listener with id = sampleListener
```

Where: sampleListener is the HTTP listener created.

**EXIT STATUS**

0 command executed successfully

1 error in executing the command

**INTERFACE EQUIVALENT**

HTTP Server folder, HTTP Listener page

**SEE ALSO**

`asadmin-delete-http-listener(IAS), asadmin-list-http-listeners(IAS)`
asadmin-create-http-qos(1AS)

NAME

asadmin-create-http-qos, create-http-qos – creates a new quality of service parameter for the named instance

SYNOPSIS


DESCRIPTION

Adds a new quality of service parameter associated with the named server instance.

OPTIONS

--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--virtualserver
virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

--bwlimit
maximum bandwidth limit, for the virtual server class or virtual server, in bytes per second. The default is no limit.

--enforcebwlimit
determines whether the bandwidth limit should be enforced or not.

--connlimit
maximum number of concurrent connections for the server, virtual server class, or virtual server.

--enforceconnlimit
determines whether the connection limit should be enforced or not.

OPERANDS

instance_name
name of the instance.

EXAMPLES

EXAMPLE 1 Using create-http-qos

asadmin> create-http-qos --user admin --password adminadmin --host fuyako --port 7070 --bwlimit 10 --enforcebwlimit=false --connlimit 2 --enforceconnlimit=true --virtualserver server1 server1

Created HTTP QOS
EXAMPLE 1 Using create-http-qos (Continued)

Where: the HTTP QOS is created for the virtual server `server1` with the instance name of `server1`.

**EXIT STATUS**

- 0: command executed successfully
- 1: error in executing the command

**INTERFACE EQUIVALENT**

Server instance, HTTP Server Virtual Servers, Instance QOS page for the server instance

**SEE ALSO**

`asadmin-delete-http-qos(1AS)`
asadmin-create-iiop-listener(1AS)

NAME

asadmin-create-iiop-listener, create-iiop-listener – adds the IIOP listener for the named instance

SYNOPSIS

create-iiop-listener --user admin_user[ --password admin_password][ --host localhost][ --port 4848][ --passwordfile filename][ --secure] [--instance instance_name] --listeneraddress address[ --iiopport iiop_port][ --enabled=true] [--property (name=value) [name=value]] listener_ID

DESCRIPTION

Adds the IIOP listener associated with the named server instance.

OPTIONS

--user

administrative user associated for the instance.

--password

administrative password corresponding to the administrative user.

--host

host name of the machine hosting the administrative instance.

--port

administrative port number associated with the administrative host.

--passwordfile

file containing passwords appropriate for the command (e.g., administrative instance).

--secure

if true, uses SSL/TLS to communicate with the administrative instance.

--instance

name of the instance.

--listeneraddress

can be the IP address or the hostname

--iiopport

IIOP port number.

--enable

determines whether the resource is enabled at runtime or not.

--property

optional attributes name/value pairs for configuring the resource.

OPERANDS

listener_ID

unique identifier for the IIOP listener to be created.

EXAMPLES

EXAMPLE 1 Using create-iiop-listener

asadmin> create-iiop-listener --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --listeneraddress 192.168.1.100 --iiopport 8080 sample_iiop_listener

Created IIOP listener with id = sample_iiop_listener

Where: sample_iiop_listener is the IIOP listener created.

EXIT STATUS

0

command executed successfully

1

error in executing the command
asadmin-create-iiop-listener(1AS)

<table>
<thead>
<tr>
<th>INTERFACE EQUIVALENT</th>
<th>ORB folder, IIOP Listener page</th>
</tr>
</thead>
</table>

SEE ALSO
asadmin-delete-iiop-listener(1AS), asadmin-list-iiop-listeners(1AS)
asadmin-create-instance(1AS)

NAME
asadmin-create-instance, create-instance – creates an application server instance with
the specified instance name

SYNOPSIS
create-instance [--user admin_user] [-password admin_password]
[--host localhost] [--port 4848] [--sysuser sys_user]
[--domain domain_name] [--local=false] [-passwordfile filename]
[--secure|-s] --instanceport instanceport instance_name

DESCRIPTION
You can create a new instance on a local or remote machine. If on the remote machine
an administration server is already running for the specified hostname, then the
system defaults to the local hostname.

To create the instance locally, not requiring the administration server to be up and
running, specify the --local option.

The named instance must not exist within that domain.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the
administrative user.
--host host name of the machine hosting the administrative
instance.
--port administrative port number associated with the
administrative host.
--sysuser owner of the domain directory.
--domain name of the domain.
--local determines if the command should delegate the request
to administrative instance or run locally.
--passwordfile file containing passwords appropriate for the command
(e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the
administrative instance.
--instanceport port where the instance listens for requests.

OPERANDS
instance_name name of the instance to be created.

EXAMPLES
EXAMPLE 1 Using create-instance in local mode
asadmin> create-instance --domain domain1 --instanceport 8967
      --sysuser adminuser server4
Created Instance server4 successfully

Where: the server4 instance is created under the domain1 domain.
asadmin-create-instance(1AS)

**EXAMPLE 2** Using create-instance in remote mode

```bash
asadmin> create-instance --sysuser adminuser --user admin --password adminadmin --host localhost --port 4848 --instanceport 8967 server4
Created Instance server4 successfully
```

Where: the server4 instance is created on the remote server for the associated user, password, host, and port.

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

Application Server instances page

**SEE ALSO**

asadmin-delete-instance(1AS), asadmin-start-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS)
### NAME
asadmin-create-javamail-resource, create-javamail-resource – registers the Javamail resource to the named instance

### SYNOPSIS
```
create-javamail-resource --user admin_user [--password admin_password]
    [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] --mailhost hostname
    --mailuser username --fromaddress address [--storeprotocol imap] [--storeprotocolclass com.sun.mail.imap.IMAPStore]
    [--transprotocol=smtp] [--transprotocolclass=com.sun.mail.smtp.SMTPTransport]
    [--debug=false] [--enabled=true] [--description text] [--property (name=value) [name=value] •] jndi_name
```

### DESCRIPTION
Registers the Javamail resource to the named instance.

### OPTIONS
- **--user**: administrative user associated for the instance.
- **--password**: administrative password corresponding to the administrative user.
- **--host**: host name of the machine hosting the administrative instance.
- **--port**: administrative port number associated with the administrative instance.
- **--passwordfile**: file containing passwords appropriate for the command (e.g., administrative instance).
- **--secure**: if true, uses SSL/TLS to communicate with the administrative instance.
- **--instance**: name of the instance.
- **--mailhost**: mail server host.
- **--mailuser**: mail account user name.
- **--fromaddress**: email address.
- **--storeprotocol**: mail server stored protocol.
- **--storeprotocolclass**: mail server stored protocol class name.
- **--transprotocol**: mail server transport protocol.
- **--transprotocolclass**: mail server transport protocol class name.
- **--debug**: if set to true, server startup in debug mode for this resource.
- **--enable**: determines whether the resource is enabled at runtime or not.
- **--description**: text description of the JDBC connection pool.
asadmin-create-javamail-resource(1AS)

--property optional attributes name/value pairs for configuring the resource.

OPERANDS

jndi_name JNDI name of the Javamail resource to be created.

EXAMPLES

EXAMPLE 1 Using create-javamail-resource

```
asadmin> create-javamail-resource --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 --mailhost localhost --mailuser sample
--fromaddress sample\@sun\com --storeprotocol imap --storeprotocolclass
com.sun.mail.imap.IMAPStore --transprotocol smtp --transprotocolclass
com.sun.mail.smtp.SMTPTransport sample_javamail_resource
```

Created the JavaMail resource with jndiname = sample_javamail_resource

Where: sample_javamail_resource is the javamail resource created. The escape character (\) is used in the fromaddress option to distinguish the dot (.) and @ sign.

EXIT STATUS

0 command executed successfully

1 error in executing the command

INTERFACE EQUIVALENT

Javamail Sessions page

SEE ALSO

asadmin-delete-javamail-resource(1AS), asadmin-list-javamail-resources(1AS)
asadmin-create-jdbc-connection-pool

NAME
asadmin-create-jdbc-connection-pool, create-jdbc-connection-pool -- registers the JDBC connection pool to the named instance

SYNOPSIS
create-jdbc-connection-pool --user admin_user[--password admin_password] [--host localhost][--port 4848][--passwordfile filename][--secure|-s][--instance instance_name][--datasourceclassname classname][--restype res_type][--steadypoolsize 8][--maxpoolsize 32][--maxwait 6000][--poolresize 2][--idletimeout 300][--isolationlevel isolation_level][--isisolationguaranteed=true][--isconnectvalidatereq=false][--validationmethod auto-commit][--validationtable table_name][--property (name=value)][--failconnection=false][--description text][--property (name=value)*][connection_pool_ID

DESCRIPTION
Registers the JDBC connection pool to the named instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative instance.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--datasourceclassname name of the vendor supplied JDBC datasource resource manager.
--restype must be specified to disambiguate when a Datasource class implements both interfaces. An error is produced when this option has a legal value and the indicated interface is not implemented by the datasource class. This option has no default value.
--steadypoolsize minimum and initial number of connections maintained in the pool.
--maxpoolsize maximum number of connections that can be created.
--maxwait the amount of time a caller will wait before a connection timeout is sent. The default is 60 seconds. A value of 0 forces the caller to wait indefinitely.
- - poolresize
  number of connections to be removed when idletimeout timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches steadypoolsize, the connection removal stops.

- - idletimeout
  maximum time (in seconds) that a connection can remain idle in the pool. After this time, the implementation can close this connection. It is recommended that this timeout is kept shorter than the server side timeout to prevent the accumulation of unusable connections in the application.

- - isolationlevel
  specifies the transaction-isolation-level on the pooled database connections. This option does not have a default value. If not specified, the pool operates with default isolation level provided by the JDBC driver. A desired isolation level can be set using one of the standard transaction isolation levels: read-uncommitted, read-committed, repeatable-read, serializable. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool. This could lead to program errors.

- - isconnectvalidatereq
  if set to true connections are validated (checked to see if they are usable) before giving out the application. The default is false.

- - validationmethod
  name of the validation table used to perform a query to validate a connection.

- - validationtable
  name of the validation table used to perform a query to validate a connection. This parameter is mandatory if connection-validation-type is set to table. Verification by accessing a user specified table may become necessary for connection validation.

- - failconnection
  if set to true, all connection in the pool must be closed if a single validation check fails; defaults to false. One attempt is made to re-establish failed connections.

- - description
  text description of the JDBC connection pool.

- - property
  optional attributes name/value pairs for configuring the connection pool.

OPERANDS
  connection_pool_id
  name of the JDBC connection pool to be created.

EXAMPLES
  EXAMPLE 1 Using create-jdbc-connection-pool
  asadmin> create-jdbc-connection-pool --user admin --password adminadmin
  --host fuyako --port 7070 --instance server1 --datasourceclassname XA
EXAMPLE 1 Using create-jdbc-connection-pool (Continued)

--restype jax.sql.DataSource --isolationlevel serializable --isconnectvalidatereq=true
--validationmethod auto-commit --description "XA Connection"
--property DatabaseName="jdbc\:pointbase\:server\:/\/@localhost/\sample"
:User=public:Password=public XA_connection_pool
Created the JDBC connection pool resource with id=XA_connection_pool

Where: the XA_connection_pool is created. The escape character "\" is used in the
--property option to distinguish the colons (:) and the backslash (/).

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
JDBC folder, Connection Pool page

SEE ALSO
asadmin-delete-jdbc-connection-pool(1AS),
asadmin-list-jdbc-connection-pools(1AS)
asadmin-create-jdbc-resource(1AS)

NAME
asadmin-create-jdbc-resource, create-jdbc-resource – registers the JDBC resource to the named instance

SYNOPSIS

DESCRIPTION
Registers the JDBC resource to the named instance.

OPTIONS
- --user administrative user associated for the instance.
- --password administrative password corresponding to the administrative user.
- --host host name of the machine hosting the administrative instance.
- --port administrative port number associated with the administrative host.
- --passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
- --secure if true, uses SSL/TLS to communicate with the administrative instance.
- --instance name of the instance.
- --connectionpoolid name of the JDBC connection pool. If two or more JDBC resource elements point to the same connection pool element, they will use the same pool connections at runtime.
- --enable determines whether the resource is enabled at runtime or not.
- --description text description of the JDBC connection pool.

OPERANDS
jndi_name JNDI name of the JDBC resource to be created.

EXAMPLES
EXAMPLE 1 Using the create-jdbc-resource command
asadmin> create-jdbc-resource --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 --connectionpoolid XA_connection_pool
--description "creating a sample jdbc resource" sample_jdbc_resource
Created the external JDBC resource with jndiname = sample_jdbc_resource

Where: sample_jdbc_resource is the resource that is created.

EXIT STATUS
0 command executed successfully
1 error in executing the command
asadmin-create-jdbc-resource(1AS)

INTERFACE
EQUIVALENT
SEE ALSO

JDBC folder, Datasource page

asadmin-delete-jdbc-resource(1AS), asadmin-list-jdbc-resources(1AS)
asadmin-create-jmsdest

**NAME**
asadmin-create-jmsdest, create-jmsdest – adds the named destination

**SYNOPSIS**
create-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] --desttype type [--property (name=value) |:name=value] *dest_name

**DESCRIPTION**
Adds the named destination.

**OPTIONS**
- **--user**
  administrative user associated for the instance.
- **--password**
  administrative password corresponding to the administrative user.
- **--host**
  host name of the machine hosting the administrative instance.
- **--port**
  administrative port number associated with the administrative host.
- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).
- **--secure**
  if true, uses SSL/TLS to communicate with the administrative instance.
- **--instance**
  name of the instance.
- **--desttype**
  type of JMS destination. Valid values are topic, and queue.
- **--property**
  name/value pairs used for specifying iMQ specific attributes to further customize the destination to be created.

**OPERANDS**
**dest_name**
name of the JMS destination. Valid value is any name that can be a Java identifier.

**EXAMPLES**
**EXAMPLE 1** Using create-jmsdest

```
asadmin> create-jmsdest --user admin --passwordfile passwords.txt
--host localhost --port 4848 --instance server1 --desttype topic
--property User=public:Password=public topic_dest
Created the JMS Destination with destype=topic
```

**EXIT STATUS**
0
command executed successfully
1
error in executing the command

**INTERFACE EQUIVALENT**

JMS Destination pages

**SEE ALSO**
asadmin-delete-jmsdest(1AS), asadmin-list-jmsdest(1AS)
asadmin-create-jms-resource(as)

NAME
asadmin-create-jms-resource, create-jms-resource – registers the JMS resource to the named instance

SYNOPSIS
create-jms-resource --user admin_user [--password admin_password]
[--host localhost] [--port 4848] [--passwordfile_filename] [--secure|--s] [--instance instance_name] --resourcetype type|--enabled=true|--description text] [--property (name=value) [name=value]*] jndi_name

DESCRIPTION
Registers the JMS resource to the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

--resourcetype

--enabled
determines whether the resource is enabled at runtime or not.

--description
text description of the JDBC connection pool.

--property
optional attributes name/value pairs for configuring the JMS resource.

OPERANDS
jndi_name
JNDI name of the JMS resource to be created.

EXAMPLE
1 Using the create-jms-resource command
asadmin> create-jms-resource --user admin --password adminadmin
--host fuyako --prot 7070 --instance server1 --resourcetype javax.jms.Queue
--description "this is a sample of creating a jms resource"
--property imqDestinationName=SimpleMessageMDB sample_jms_resource
Created the JMS resource with jndiname = sample_jms_resource

Where: the sample_jms_resource is the resource that is created.
<table>
<thead>
<tr>
<th>asadmin-create-jms-resource(1AS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXIT STATUS</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

JMS folder, Destinations page

**SEE ALSO**
asadmin-delete-jms-resource(1AS), asadmin-list-jms-resources(1AS)
asadmin-create-jndi-resource(1AS)

| NAME | asadmin-create-jndi-resource, create-jndi-resource – registers the JNDI resource to the named instance |
| SYNOPSIS | create-jndi-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|--ssl] [--instance instance_name] --jndilookupname lookup_name --resourcetype type --factoryclass class_name [--enabled=true] [--description text] [--property (name=value) [name=value]*] jndi_name |
| DESCRIPTION | Registers the JNDI resource to the named instance. |
| OPTIONS | --user administrative user associated for the instance. --password administrative password corresponding to the administrative user. --host host name of the machine hosting the administrative instance. --port administrative port number associated with the administrative host. --passwordfile file containing passwords appropriate for the command (e.g., administrative instance). --secure if true, uses SSL/TLS to communicate with the administrative instance. --instance name of the instance. --jndilookupname lookup name used by external container. --resourcetype JNDI resource type which can be: topic or queue. --factoryclass class that creates the JNDI resource. --enabled determines whether the resource is enabled at runtime or not. --description text description of the JDBC connection pool. --property optional attributes name/value pairs for configuring the JNDI resource. |
| OPERANDS | jndi_name name of the JNDI resource to be created. |
| EXAMPLES | **EXAMPLE 1** Using the create-jndi-resource command

```
asadmin> create-jndi-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --jndilookupname sample_jndi --resourcetype queue --factoryclass sampleClass --description "this is a sample jndi resource" sample_jndi_resource
Created the JNDI resource with jndiname = sample_jndi_resource
```

Where: sample_jndi_resource is the JNDI resource created.
<table>
<thead>
<tr>
<th>EXIT STATUS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**JNDI folder, External page**

**SEE ALSO**

asadmin-delete-jndi-resource(1AS), asadmin-list-jndi-resources(1AS)
asadmin-create-jvm-options(1AS)

NAME
asadmin-create-jvm-options, create-jvm-options – creates the JVM options from the Java configuration or profiler elements

SYNOPSIS
create-jvm-options --user admin_user[ --password admin_password][ --host localhost][ --port 4848][ --passwordfile filename][ --secure][--instance instance_name][ --profiler=false][:jvm_option_name=jvm_option_value]*

DESCRIPTION
Creates the JVM options in the Java configuration or Profiler elements of the server.xml file. You can enter more than one JVM option separated by a colon (:). If the JVM option starts with a dash (-) then use two dashes (--) before the operand to distinguish that JVM option is an operand and not an option.

JVM options are used to record the settings needed to get a particular profiler going.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true.

OPERANDS
jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value.

EXAMPLES
EXAMPLE 1 Using create-jvm-options
```
 asadmin> create-jvm-options --user admin --password adminadmin
 --host fuyako --port 7070 --instance server1 --profiler=false
 --*-DDdebug=true*:*-Xmx256m*:*-Dcom.sun.aas.imqBin=*\export/as7se/imq\bin*
 JVM options created
```

Where the JVM options are created. The double dash (--) is used between --profiler options and the operand because – indicated the end of the options and the following text is the operand. The double dash (--) is necessary here since there are single dashes (i.e., --DDebug) in the operand. To distinguish between the options and the operand, the double dash (--) is used.
### asadmin-create-jvm-options(1AS)

<table>
<thead>
<tr>
<th>EXIT STATUS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

### SEE ALSO
- asadmin-delete-jvm-options(1AS)
asadmin-create-lifecycle-module

NAME
asadmin-create-lifecycle-module, create-lifecycle-module – adds a lifecycle module for the named instance

SYNOPSIS
create-lifecycle-module --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename][--secure|-s] [--instance instance_name] --classname class_name[--classpath classpath][--loadorder load_order][--failurerecover failure_fatal] [--enabled=true] [--property (name=value) [*] module_name

DESCRIPTION
Creates the lifecycle module associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--classname fully qualified name of the startup class.
--classpath indicates where this module is actually located if it is not under applications-root.
--loadorder an integer value that can be used to force the order in which deployed lifecycle modules are loaded at server startup. Smaller numbered modules get loaded sooner. Order is unspecified if two or more lifecycle modules have the same load-order value.
--failurerecover if true indicates abort server startup if this module does not load properly.
--enabled determines whether the resource is enabled at runtime or not.
--description text description of the resource.
--property optional attributes name/value pairs for configuring the resource.

OPERANDS
module_name unique identifier for the deployed server lifecycle event listener module.
asadmin-create-lifecycle-module(1AS)

EXAMPLES

**EXAMPLE 1** using create-lifecycle-module

```bash
asadmin> create-lifecycle-module --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 --classname "com.acme.CustomSetup"
--classpath "/export/customSetup" --loadorder 1 --failurefatal=true
--description "this is a sample customSetup" --property rmi=Server="acme1:7070"
:timeout=30 customSetup

Created the Lifecycle module with module name = customSetup
```

Where: customSetup is the lifecycle module created. The escape character (\) is used in the property option to distinguish the colons (:).

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

Application Lifecycle Modules page

**SEE ALSO**

asadmin-delete-lifecycle-module(1AS),
asadmin-list-lifecycle-modules(1AS)
asadmin-create-mime(1AS)

| NAME | asadmin-create-mime, create-mime – adds the MIME type for the named instance |
| SYNOPSIS | `create-mime --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--instance instance_name] --mimefile filename mime_ID` |
| DESCRIPTION | Adds the MIME type associated with the named server instance. The server determines the MIME type of a requested resource by invoking the type-by-extension directive in the ObjectType section of the obj.conf file. The type-by-extension function does not work if no MIME element has been defined in the server element. |
| OPTIONS | `-user` administrative user associated for the instance.  
`--password` administrative password corresponding to the administrative user.  
`--host` host name of the machine hosting the administrative instance.  
`--port` administrative port number associated with the administrative host.  
`--passwordfile` file containing passwords appropriate for the command (e.g., administrative instance).  
`--secure` if true, uses SSL/TLS to communicate with the administrative instance.  
`--instance` name of the instance.  
`--mimefile` name of a MIME types file. |
| OPERANDS | `mime_id` internal name for the MIME types listing. It is used in a virtual-server element to define the MIME types used by the virtual server. |
| EXAMPLES | **EXAMPLE 1** Using create-mime  
asadmin> create-mime --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --mimefile mime.types  
sampleMIME  
Created Mime with id = sampleMIME  

Where: sampleMIME is the name of the MIME created. |
| EXIT STATUS | 0 command executed successfully  
1 error in executing the command |
| INTERFACE EQUIVALENT SEE ALSO | HTTP Server node, MIME Type Files page  
asadmin-delete-mime(1AS), asadmin-list-mimes(1AS) |
asadmin-create-persistence-resource(1AS)

NAME
asadmin-create-persistence-resource, create-persistence-resource – registers the persistence resource to the named instance

SYNOPSIS

DESCRIPTION
Registers the persistence resource associated with the specified JNDI name from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

--jdbcjndiname
JDBC resource with which database connections are obtained. Must be the name of one of the pre-created JDBC resources.

--factoryclass
class that creates persistence manager instance.

--enable
determines whether the resource is enabled at runtime or not.

--description
text description of the resource.

--property
optional attributes name/value pairs for configuring the resource.

OPERANDS
jndi_name
JNDI name of the persistence manager factory resource.

EXAMPLES
EXAMPLE 1 Using create-persistence-resource
asadmin> create-persistence-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --jdbcjndiname sample_jndi_resource --factoryclass "com.pmf.class" sample_persistence_resource
Created Persistence manager resource with jndiname = sample_persistence_resource
EXAMPLE 1 Using create-persistence-resource (Continued)

Where: `sample_persistence_resource` is the persistence manager factory resource created.

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

Persistence Manager page

**SEE ALSO**

`delete-persistence-resource(1AS), list-persistence-resources(1AS)`
asadmin-create-profiler(1AS)

NAME
asadmin-create-profiler, create-profiler – creates the profiler element

SYNOPSIS
create-profiler --user admin_user[ --password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure|-s][--instance instance_name] --classpath classpath[--nativelibpath native_library_path][--enabled=true] [--property (name=value) [name=value]*] profiler_name

DESCRIPTION
Creates the profiler element. A server instance is tied to a particular profiler, by the profiler element in the Java configuration. Changing a profiler requires you to restart the server.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--classpath Java classpath string that specifies the classes needed by the profiler.
--nativelibpath automatically constructed to be a concatenation of the Application Server installation relative path for its native shared libraries, standard JRE native library path, the shell environment setting (LD_LIBRARY_PATH on UNIX) and any path that may be specified in the profile element.
--enabled profiler is enabled by default.
--property name/value pairs of provider specific attributes.

OPERANDS
profiler_name name of the profiler.

EXAMPLES
Example 1 Using create-profiler

asadmin> create-profiler --user admin --passwordfile passwords.txt
--host localhost --port 4848 --instance server1 --classpath com.iplanet.ias.profile.Profiler
--nativelibpath /u/home/lib --no-enabled --property defaultuser=admin:password=adminadmin
sample_profiler
Created Profiler with id = sample_profiler
EXAMPLE 1 Using create-profiler (Continued)

Where: sample_profiler is the profiler created.

EXIT STATUS

0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT

Application Server instances, JVM Settings tab

SEE ALSO

asadmin-delete-profiler(1AS) asadmin-list-profilers(1AS)
# NAME
asadmin-create-ssl, create-ssl – creates the SSL element in the HTTP listener or IIOP listener.

# SYNOPSIS
```
create-ssl --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure|-s][--type http-listener|iiop-listener|iiop-service] --certname cert_name[--instance instance_name][--ssl2enabled=false][--ssl2ciphers ssl_2_ciphers][--ssl3enabled=true][--ssl3tlscedifiers ssl3_tls_ciphers] [--tlsenabled=true] [--tlsrollbackenabled=true][--clientauthenabled=false] [listener_id]
```

# DESCRIPTION
Deletes the ssl element from the HTTP listener or IIOP listener.

## OPTIONS
- **--user**
  administrative user associated for the instance.

- **--password**
  administrative password corresponding to the administrative user.

- **--host**
  host name of the machine hosting the administrative instance.

- **--port**
  administrative port number associated with the administrative host.

- **--secure**
  indicates communication with the administrative instance in secured mode.

- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).

- **--instance**
  name of the instance.

- **--type**
  type of service or listener that the SSL is created for. The type can be: http-listener, iiop-listener, and iiop-service.

- **--certname**
  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 in this attribute is optional.

- **--ssl2enabled**
  determines whether SSL2 is enabled.

- **--ssl2ciphers**
  a comma separated list of the SSL2 ciphers used. Use the prefix + to enable or — to disable. Allowed values are: rc4, rc4export, rc2, rc2export, idea, des, desede3. If no value is specified, all supported ciphers are assumed to be enabled.

- **--ssl3enabled**
  determines whether SSL3 is enabled.

- **--ssl3ciphers**
  a comma separated list of the SSL3 ciphers used. Use the prefix + to enable or — to disable. Allowed values
asadmin-create-ssl(1AS)

```
arc rsa_rc4_128_md5, rsa3des_sha,
    rsa_des_sha, rsa_rc4 40_md5, rsa_rc2 40_md5,
    rsa_null_md5. Allowed TSL values are:
    rsa_des_56_sha, rsa_rc4_56_sha. If no value is
    specified, all supported ciphers are assumed to be
    enabled.

--tlsenabled determines whether TLS is enabled.
--tlsrollbackenabled determines whether TLS rollback is enabled. TLS
    rollback should be enabled for Microsoft Internet
    Explorer 5.0 and 5.5.
--clientauthenabled determines whether SSL3 client authentication is
    performed on every request independent of ACL-based
    access control.

OPERANDS

  listener_ID the ID of the listener or service that the SSL is created
  for.

EXAMPLES

  EXAMPLE 1 Using create-ssl

  asadmin> create-ssl --user admin --password adminadmin

  --host fuyako --port 7070 --type http-listener --certname sampleCert

  --instance server1 --ssl2enabled=true --ssl2ciphers rc4,rc2,des

  --ssl3enabled=false --ssl3ciphers rsa_rc4_128_md5,rsa3des_sha,rsa_des_sha,

  rsa_rc4_40_md5 --tlsenabled=false --tlsrollbackenabled=false

  --clientauthenabled=false http-listener-1

  Created SSL in HTTP Listener

  Where: SSL is created for http-listener-1.

EXIT STATUS

  0 command executed successfully

  1 error in executing the command

INTERFACE

  HTTP Server folder, HTTP Listeners page, ORB folder, IIOP Listeners page

EQUIVALENT

  SEE ALSO

  asadmin-delete-ssl(1AS)
NAME
asadmin-create-virtual-server, create-virtual-server – adds the named virtual server

SYNOPSIS
create-virtual-server --user admin_user [--password admin_password]
                    [--host localhost] [--port 4848] [--passwordfile filename]
                    [--secure|-s] [--instance instance_name] [--hosts hosts]
                    ------ mime_types_file [--htmlchemy [http_listeners]]
                    [--defaultwebmodule default_web_module]
                    [--configfile config_file] [--defaultobj default_object]
                    [--state on] [--acls acls] [--acceptlang=false]
                    [--logfile log_file] [--property (name=value)]
                    [name=value]*]
virtual_server_ID

DESCRIPTION
Creates the named virtual server. Virtualization in the Application Server allows
multiple URL domains to be served by the same HTTP server process which is
listening on multiple host addresses. If the application is available at two virtual
servers, they still share the same physical resource pools.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the
    administrative user.
--host host name of the machine hosting the administrative
    instance.
--port administrative port number associated with the
    administrative host.
--passwordfile file containing passwords appropriate for the command
    (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the
    administrative instance.
--instance name of the instance.
--hosts a comma separated (,) list of values allowed in the host
    request header to select the current virtual server. Each
    virtual server that is configured to the same connection
    group must have a unique hosts value for that group.
--mime the ID of the mime element used by the virtual server.
--httplisteners optional; a comma separated (,) list of HTTP listener
    IDs. Required only for a virtual server that is not the
    default virtual server.
--defaultwebmodule standalone web module associated with this virtual
    server by default.
--configfile typically all virtual server initialization is from
    $INSTANCE_ROOT/config/obj.conf. This can be changed using this attribute.
--defaultobj
names the object loaded from an obj.conf file which is
default. The default object is expected to have all the
name translation directives for the virtual server. Any
server behavior that is configured in the default object
affects the entire virtual server class.

--state
determines whether a virtual server is active (on) or
inactive (off or disabled). Default is active (on). When
inactive, the virtual server does not service requests.

--acls
a comma-separated list of ID attributes of ACL
elements. Specifies the ACL files used by the virtual
server.

--acceptlang
when turned on, the server parses the
Accept-Language header and sends an appropriate
language version based on which language the client
can accept. Set this value to ON only if the server
supports multiple languages. The default setting is
determined from the virtual-server-class.

--logfile
name of the file where the log has to be written to.

--property
optional attributes name/value pairs for configuring
the connection pool.

OPERANDS
virtual_server_id
identifies the unique ID for the virtual server to be
created. This virtual server ID cannot begin with a
number.

EXAMPLES
EXAMPLE 1 Using create-virtual-server
asadmin> create-virtual-server --user admin --password adminadmin
--host fuyako --port 7070 --httplisteners http-listener-1 --defaultwebmodule simple
--configfile config/obj.conf --defaultobj default --state on --acls acl1
--no-acceptlang --logfile server.log --property User=admin:Password=admin
--hosts sample1,sample2 --mime mime1 sample_vs1
Created virtual server with id = sample_vs1

Where sample_vs1 is the virtual server created.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
HTTP Server node, Virtual Servers page

EQUIVALENT
SEE ALSO
asadmin-delete-virtual-server(1AS),
asadmin-list-virtual-servers(1AS)
asadmin-delete-acl(1AS)

NAME
asadmin-delete-acl, delete-acl – removes the access control list file for the named instance

SYNOPSIS
delete-acl --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure|-s][--instance instance_name] acl_ID

DESCRIPTION
Gets the access control lists associated with the named server instance.

OPTIONS
- --user administrative user associated for the instance.
- --password administrative password corresponding to the administrative user.
- --host host name of the machine hosting the administrative instance.
- --port administrative port number associated with the administrative host.
- --secure indicates communication with the administrative instance in secured mode.
- --passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
- --instance name of the instance.

OPERANDS
acl_ID internal name for the ACL file listing. This ID is used in a virtual server element to define the ACL file used by the virtual server.

EXAMPLES
EXAMPLE 1 Using delete-acl
asadmin> delete-acl --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 sampleACL
Deleted ACL with id = sampleACL

Where: sampleACL is the ACL that is deleted.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Access Control List page

SEE ALSO
asadmin-create-acl(1AS), asadmin-list-acls(1AS)
asadmin-delete-authdb(1AS)

NAME
asadmin-delete-authdb, delete-authdb – removes the authorized database for the named instance

SYNOPSIS

DESCRIPTION
Removes the authorized database associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--virtualserver virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

OPERANDS
authdb_id user database name in the virtual server’s ACL file.

EXAMPLES
EXAMPLE 1 Using delete-authdb
asadmin delete-authdb --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --virtualserver server1 sampleAuth
Deleted AuthDB with id = sampleAuth

Where: sampleAuth is the authdb deleted.

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-create-authdb(1AS), asadmin-list-authdbs(1AS)
NAME
asadmin-delete-auth-realm, delete-auth-realm – removes the named authorized realm

SYNOPSIS
delete-auth-realm --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure|-s][--instance instance_name] auth_realm_name

DESCRIPTION
Removes the named authorized realm.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
auth_realm_name
name of this realm.

EXAMPLES
EXAMPLE 1 Using delete-auth-realm
asadmin> delete-auth-realm --user admin --passwordfile passwords.txt
--host localhost --port 4848 --instance server1 db
Deleted Auth realm with id = db

Where db is the auth realm deleted.

EXIT STATUS
0  command executed successfully
1  error in executing the command

SEE ALSO
asadmin-create-auth-realm(1), asadmin-list-auth-realms(1)
asadmin-delete-custom-resource(1AS)

NAME
asadmin-delete-custom-resource, delete-custom-resource – removes the custom resource from the named instance

SYNOPSIS
delete-custom-resource --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename] [--secure|-s] [--instance instance_name] jndi_name

DESCRIPTION
Removes the custom resource from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
jndi_name
JNDI name of the custom resource to be deleted.

EXAMPLES
EXAMPLE 1 Using delete-custom-resource
asadmin> delete-custom-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_custom_resource
Deleted the custom resource with jndiname = sample_custom_resource

Where sample_custom_resource is the custom resource deleted.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
JNDI folder, Custom page

EQUIVALENT
SEE ALSO
asadmin-create-custom-resource(1AS),
asadmin-list-custom-resources(1AS)
asadmin-delete-domain(1AS)

NAME  asadmin-delete-domain, delete-domain – deletes the given domain

SYNOPSIS delete-domain domain_name

DESCRIPTION Use the delete-domain command to delete the specified domain. The domain must already exist, but the instances within the domain must not be executing. The delete-domain command can be run locally only.

OPTIONS domain_name name of the domain; must be a unique name.

EXAMPLES EXAMPLE 1 Using delete-domain

asadmin> delete-domain domain1
deleted domain domain1 successfully

Where: the domain1 domain is deleted.

EXIT STATUS

0 command executed successfully
1 error in executing the command

SEE ALSO asadmin-create-domain(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS), asadmin-list-domains(1AS), asadmin-multimode(1AS)
asadmin-delete-file-user(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-delete-file-user, delete-file-user – removes the named file user</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>delete-file-user --user admin_user[--password admin_password] [--host localhost] [--port 4848][--passwordfile filename][--secure</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Removes the named file user.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--user</td>
</tr>
<tr>
<td></td>
<td>--password</td>
</tr>
<tr>
<td></td>
<td>--host</td>
</tr>
<tr>
<td></td>
<td>--port</td>
</tr>
<tr>
<td></td>
<td>--passwordfile</td>
</tr>
<tr>
<td></td>
<td>--secure</td>
</tr>
<tr>
<td></td>
<td>--instance</td>
</tr>
<tr>
<td>OPERANDS</td>
<td>user_name</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using the delete-file-user command</td>
</tr>
</tbody>
</table>
| | asadmin> delete-file-user --user admin --password admin_password
| | --host fuyako --port 7070 --instance server1 sample_user |
| | Deleted File user sample_user |
| | Where: the sample_user is the file user deleted. |
| EXIT STATUS | 0 | command executed successfully |
| | 1 | error in executing the command |
| SEE ALSO | asadmin-create-file-user(1AS), asadmin-list-file-users(1AS), asadmin-update-file-user(1AS), asadmin-list-file-groups(1AS) |
asadmin-delete-http-listener(1AS)

NAME
asadmin-delete-http-listener, delete-http-listener - removes the HTTP listener for the named instance

SYNOPSIS
delete-http-listener --user admin_user [--password admin_password]
            [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
            [--instance instance_name] httpListener_ID

DESCRIPTION
Removes the HTTP listeners associated with the named server instance.

OPTIONS
--user      administrative user associated for the instance.
--password  administrative password corresponding to the administrative user.
--host      host name of the machine hosting the administrative instance.
--port      administrative port number associated with the administrative host.
--passwordfile  file containing passwords appropriate for the command (e.g., administrative instance).
--secure    if true, uses SSL/TLS to communicate with the administrative instance.
--instance  name of the instance.

OPERANDS
listener_id  listener ID of the HTTP listener.

EXAMPLES
EXAMPLE 1 Using delete-http-listener
asadmin> delete-http-listener --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 sampleListener
Deleted HTTP listener with id = sampleListener

Where: sampleListener is the HTTP listener deleted.

EXIT STATUS
0  command executed successfully
1  error in executing the command

INTERFACE
HTTP Server folder, HTTP Listener page

EQUIVALENT
SEE ALSO
asadmin-create-http-listener(1AS), asadmin-list-http-listeners(1AS)
asadmin-delete-http-qos(1AS)

NAME
asadmin-delete-http-qos, delete-http-qos – removes the quality of service parameter for the named instance

SYNOPSIS

DESCRIPTION
Removes the quality of service parameter associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--virtualserver virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using delete-http-qos
asadmin> delete-http-qos --user admin --password adminadmin --host fuyako --port 7070 --virtualserver server1 server1
Deleted HTTP QOS with id = server1

Where: HTTP QOS is deleted for virtual server server1 and instance name server1.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Server instance, HTTP Server Virtual Servers, Instance QOS page for the server instance

SEE ALSO
asadmin-create-http-qos(1AS)
asadmin-delete-iiop-listener(1AS)

NAME
asadmin-delete-iiop-listener, delete-iiop-listener – removes the IIOP listener for the named instance

SYNOPSIS

DESCRIPTION
Removes the IIOP listener associated with the named server instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
listener_id
unique identifier for the IIOP listener to be deleted.

EXAMPLES
EXAMPLE 1 Using delete-iiop-listener
asadmin> delete-iiop-listener --user admin --password adminadmin
--host fuyako --port 7070 sample_iiop_listener
Deleted IIOP listener with id = sample_iiop_listener

Where: sample_iiop_listener is the IIOP listener deleted.

EXIT STATUS
0
command executed successfully

1
error in executing the command

INTERFACE EQUIVALENT
ORB folder, IIOP Listener page

SEE ALSO
asadmin-create-iiop-listener(1AS), asadmin-list-iiop-listeners(1AS)
asadmin-delete-instance (1AS)

NAME
asadmin-delete-instance, delete-instance – deletes the instance that is not running

SYNOPSIS
del**ete-instance** [--user admin_user] [--password admin_password]
              [--host localhost] [--port 4848] [--local=false]
              [--domain domain_name] [--passwordfile filename]
              [--secure|-s]instance_name

DESCRIPTION
Use the delete-instance command to delete the instance that you specify. The delete-instance command can be run both locally and remotely.

To delete the instance locally, not requiring the administration server to be up and running, specify the --local option.

To delete the instance remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server.

Use this command with discretion since it is destructive and there is no undo.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--domain name of the domain.
--local determines if the command should delegate the request to administrative instance or run locally.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance to be deleted.

EXAMPLES
EXAMPLE 1 Using delete-instance in local mode
asadmin> delete-instance --domain domain1 server1
Deleted Instance server1 successfully

Where: the server1 instance for the domain1 domain is deleted on the local machine.
EXAMPLE 2 Using delete-instance in remote mode

```
admin> delete-instance --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1
Deleted Instance server1 successfully
```

Where: the server1 instance for the domain associated with the specified user, passwords in the password file, host, and port number is deleted on the remote machine.

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

Server Instance page

**SEE ALSO**

asadmin-create-instance(1AS), asadmin-start-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS)
asadmin-delete-javamail-resource(1AS)

NAME
asadmin-delete-javamail-resource, delete-javamail-resource – removes the Javamail resource from the named instance

SYNOPSIS
delete-javamail-resource --user admin_user[--password admin_password]
[--host localhost][--port 4848][--passwordfile filename]
[--secure|-s][--instance instance_name] jndi_name

DESCRIPTION
Removes the Javamail resource from the named instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.

OPERANDS
jndi_name JNDI name of the Javamail resource to be deleted.

EXAMPLES
EXAMPLE 1 Using delete-javamail-resource
asadmin> delete-javamail-resource --user admin --password admin_password
--host fuyako --port 7070 --instance server1 sample_javamail_resource
Deleted the JavaMail resource with jndiname = sample_javamail_resource

Where: sample_javamail_resource is the javamail resource deleted.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIV ALENT
Javamail Sessions page

SEE ALSO
asadmin-create-javamail-resource(1AS),
asadmin-list-javamail-resources(1AS)
asadmin-delete-jdbc-connection-pool(1AS)

NAME
asadmin-delete-jdbc-connection-pool, delete-jdbc-connection-pool – removes the JDBC connection pool from the named instance

SYNOPSIS
delete-jdbc-connection-pool --user admin_user [ --password admin_password ] [ --host localhost ] [ --port 4848 ] [ --passwordfile filename ] [ --secure | -s ] [ --instance instance_name ] connection_pool_ID

DESCRIPTION
Removes the JDBC resource from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
collection_pool_id
name of the JDBC connection pool to be created.

EXAMPLES
EXAMPLE 1 Using the delete-jdbc-connection-pool command
asadmin> delete-jdbc-connection-pool --user admin --password adminadmin
--host fuyako port 7070 --instance server1 XA_connection_pool
Deleted the JDBC connection pool resource with id = XA_connection_pool

Where: the XA_connection_pool resource is deleted.

EXIT STATUS
0
command executed successfully
1
error in executing the command

INTERFACE EQUIVALENT
JDBC folder, Connection Pool page

SEE ALSO
asadmin-create-jdbc-connection-pool(1AS),
asadmin-list-jdbc-connection-pools(1AS)
## NAME
asadmin-delete-jdbc-resource, delete-jdbc-resource – removes the JDBC resource from the named instance

## SYNOPSIS
```
delete-jdbc-resource --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure][--instance instance_name]jndi_name
```

## DESCRIPTION
Removes the JDBC resource from the named instance.

## OPTIONS
- `--user`   administrative user associated for the instance.
- `--password`   administrative password corresponding to the administrative user.
- `--host`   host name of the machine hosting the administrative instance.
- `--port`   administrative port number associated with the administrative host.
- `--passwordfile`   file containing passwords appropriate for the command (e.g., administrative instance).
- `--secure`   if true, uses SSL/TLS to communicate with the administrative instance.
- `--instance`   name of the instance.

## OPERANDS
- `jndi_name`   name of the JDBC resource to be deleted.

## EXAMPLES
**EXAMPLE 1** Using the delete-jdbc-resource command
```
asadmin> delete-jdbc-resource --user admin --password adminadmin
--host fuyako --port 7070 instance server1 sample_jdbc_resource
```
Deleted the external JDBC resource with jndiname = sample_jdbc_resource

Where: `sample_jdbc_resource` is the resource that is deleted.

## EXIT STATUS
- 0 command executed successfully
- 1 error in executing the command

## INTERFACE EQUIVALENT
JDBC folder, Datasource page

## SEE ALSO
asadmin-create-jdbc-resource(1AS), asadmin-list-jdbc-resources(1AS)
asadmin-delete-jmsdest(1AS)

NAME
asadmin-delete-jmsdest, delete-jmsdest – destroys the named destination

SYNOPSIS
delete-jmsdest --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename] [--secure|-s][--instance instance_name] --desttype type dest_name

DESCRIPTION
Destroys the named destinations.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--instance name of the instance.
--desttype type of JMS destination. Valid values are topic, and queue.

OPERANDS
dest_name name of the JMS destination. Valid value is any name that can be a Java identifier.

EXAMPLES
EXAMPLE 1 Using delete-jmsdest
asadmin> delete-jmsdest --user admin --password adminadmin --host localhost port 4848 --instance server1 --desttype topic topic_dest
Deleted the JMS Destination with desttype=topic

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
JMS Destination pages

SEE ALSO
asadmin-create-jmsdest(1AS), asadmin-list-jmsdest(1AS)
asadmin-delete-jms-resource(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-delete-jms-resource, delete-jms-resource – removes the JMS resource from the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>delete-jms-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--instance instance_name] jndi_name</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Removes the JMS resource from the named instance.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--user administrative user associated for the instance.</td>
</tr>
<tr>
<td></td>
<td>--password administrative password corresponding to the administrative user.</td>
</tr>
<tr>
<td></td>
<td>--host host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--port administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td></td>
<td>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td></td>
<td>--secure if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--instance name of the instance.</td>
</tr>
<tr>
<td>OPERANDS</td>
<td>jndi_name JNDI name of the JMS resource to be deleted.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using the delete-jms-resource command</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; delete-jms-resource --user admin --password admin_password --host fuyako --port 7070 --instance server1 sample_jms_resource</td>
</tr>
<tr>
<td></td>
<td>Deleted the JMS resource with jndiname = sample_jms_resource</td>
</tr>
<tr>
<td></td>
<td>Where: sample_jms_resource is the resource that is deleted.</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>INTERFACE EQUIVALENT</td>
<td>JMS folder, Destinations page</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-create-jms-resource(1AS), asadmin-list-jms-resources(1AS)</td>
</tr>
</tbody>
</table>
asadmin-delete-jndi-resource(1AS)

NAME
asadmin-delete-jndi-resource, delete-jndi-resource – removes the JNDI resource from the named instance

SYNOPSIS

DESCRIPTION
Removes the JNDI resource from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
jndi_name
name of the JNDI resource to be deleted.

EXAMPLES
EXAMPLE 1 Using the delete-jndi-resource command
asadmin> delete-jndi-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_jndi_resource
Created the JNDI resource with jndiname = sample_jndi_resource

Where: sample_jndi_resource is the JNDI resource to be deleted.

EXIT STATUS
0
command executed successfully
1
error in executing the command

INTERFACE
JNDI folder, External page

EQUIVALENT
SEE ALSO
asadmin-create-jndi-resource(1AS), asadmin-list-jndi-resources(1AS)
asadmin-delete-jvm-options(1AS)

NAME
asadmin-delete-jvm-options, delete-jvm-options – deletes the JVM options from the
Java configuration or profiler elements

SYNOPSIS
delete-jvm-options --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure|s] [--instance instance_name] [--profiler=false] (jvm_option_name=jvm_option_value)
[:jvm_option_name=jvm_option_value]*

DESCRIPTION
Deletes the JVM options in the Java configuration or Profiler elements of the
server.xml file. You can enter more than one JVM option separated by a colon (:) . If
the JVM option starts with a dash (-) then use two dashes (--) before the operand to
distinguish that JVM option is an operand and not an option.

JVM options are used to record the settings needed to get a particular profiler going.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the
administrative user.
--host host name of the machine hosting the administrative
instance.
--port administrative port number associated with the
administrative host.
--passwordfile file containing passwords appropriate for the command
(e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the
administrative instance.
--instance name of the instance.
--profiler indicates if the JVM options is for the profiler. Profiler
must exist for this option to be true.

OPERANDS
jvm_option_name=jvm_option_value The left side of the equal sign (=) is the JVM option
name. The right side of the equal sign (=) is the
jvm_option_value.

EXAMPLES
EXAMPLE 1 Using delete-jvm-options
asadmin> delete-jvm-options --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 --profiler=true
--"-DDebug=true":*:Xmx256m*:*:Dcom.sun.aas.imq8bin*:*/export/as7se/imq8/bin*
JVM options deleted

Where the JVM options are creaded. The double dash (—) is used between
--profiler options and the operand because – indicated the end of the options and
the following text is the operand. The double dash (—) is necessary here since there
are single dashes (i.e., —DDebug) in the operand. To distinguish between the options
and the operand, the double dash (—) is used.
asadmin-delete-jvm-options(1AS)

EXIT STATUS

0 command executed successfully
1 error in executing the command

SEE ALSO

asadmin-create-jvm-options(1AS)
asadmin-delete-lifecycle-module(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-delete-lifecycle-module, delete-lifecycle-module – removes the lifecycle module for the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>delete-lifecycle-module --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Removes the lifecycle module associated with the named server instance.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--user administrative user associated for the instance.</td>
</tr>
<tr>
<td></td>
<td>--password administrative password corresponding to the administrative user.</td>
</tr>
<tr>
<td></td>
<td>--host host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--port administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td></td>
<td>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td></td>
<td>--secure if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--instance name of the instance.</td>
</tr>
<tr>
<td>OPERANDS</td>
<td>module_name unique identifier for the deployed server lifecycle event listener module.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using delete-lifecycle-module</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; delete-lifecycle-module --user admin --password adminadmin --host fuyako --port 7070 customSetup</td>
</tr>
<tr>
<td></td>
<td>Deleted the Lifecycle module with module name = customSetup</td>
</tr>
<tr>
<td></td>
<td>Where: customSetup is the lifecycle module deleted.</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>INTERFACE EQUIVALENT</td>
<td>Application Lifecycle Modules page</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-create-lifecycle-module(1AS), asadmin-list-lifecycle-modules(1AS)</td>
</tr>
</tbody>
</table>
asadmin-delete-mime(1AS)

NAME
asadmin-delete-mime, delete-mime – removes the MIME type for the named instance

SYNOPSIS
delete-mime --user admin_user[ --password admin_password][ --host localhost][ --port 4848][ --passwordfile filename][ --secure |-s][ --instance instance_name] mime_ID

DESCRIPTION
Removes the MIME types associated with the named server instance. The server determines the MIME type of a requested resource by invoking the type-by-extension directive in the ObjectType section of the obj.conf file. The type-by-extension function does not work if no MIME element has been defined in the server element.

OPTIONS
--user
   administrative user associated for the instance.

--password
   administrative password corresponding to the administrative user.

--host
   host name of the machine hosting the administrative instance.

--port
   administrative port number associated with the administrative host.

--passwordfile
   file containing passwords appropriate for the command (e.g., administrative instance).

--secure
   if true, uses SSL/TLS to communicate with the administrative instance.

--instance
   name of the instance.

OPERANDS
mime_ID
   internal name for the MIME types listing. It is used in a virtual-server element to define the MIME types used by the virtual server.

EXAMPLES
EXAMPLE 1 Using delete-mime
asadmin> delete-mime --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 sampleMIME
Deleted Mime with id = sampleMIME

Where: sampleMIME is the name of the MIME deleted.

EXIT STATUS
0    command executed successfully
1    error in executing the command

INTERFACE EQUIVALENT
HTTP Server node, MIME Type Files page

SEE ALSO
asadmin-create-mime(1AS), asadmin-list-mimes(1AS)
asadmin-delete-persistence-resource(1AS)

NAME
asadmin-delete-persistence-resource, delete-persistence-resource – removes the persistence resource from the named instance

SYNOPSIS
delete-persistence-resource --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure|-s][--instance instance_name] jndi_name

DESCRIPTION
Removes the persistence resource associated with the specified JNDI name from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

OPERANDS
jndi_name
JNDI name of the persistence manager factory resource.

EXAMPLES
EXAMPLE 1 Using delete-persistence-resource
asadmin> delete-persistence-resource --user admin --password adminadmin
--host fuyako --port 7070 --instance server1 sample_persistence_resource
Deleted Persistence manager resource with jndiname = sample_persistence_resource

Where: sample_persistence_resource is the persistence manager factory resource to be deleted.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
Persistence Manager page

EQUIVALENT
SEE ALSO
asadmin-create-persistence-resource(1AS),
asadmin-list-persistence-resources(1AS)
asadmin-delete-profiler(1AS)

NAME
asadmin-delete-profiler, delete-profiler – deletes the profiler element

SYNOPSIS
delete-profiler --user admin_user[--password admin_password][--host localhost] [--port 4848] [--passwordfile filename][--secure][--instance_name

DESCRIPTION
Deletes the profiler element. A server instance is tied to a particular profiler by the profiler element in the Java configuration. Changing a profiler requires you to restart the server.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using delete-profiler
asadmin> delete-profiler --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1
Deleted Profiler

Where: profiler is deleted from instance server1.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Application Server Instances, JVM Settings tab

SEE ALSO
asadmin-create-profiler(1AS) asadmin-list-profilers(1AS)
asadmin-delete-ssl(1AS)

**NAME**
asadmin-delete-ssl, delete-ssl – deletes the ssl element from the HTTP listener or IIOP listener

**SYNOPSIS**
delete-ssl --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] --type [http-listener | iiop-listener | iiop-service] [--instance instance_name] [listener_id]

**DESCRIPTION**
Deletes the ssl element from the HTTP listener or IIOP listener.

**OPTIONS**
- --user
  administrative user associated for the instance.
- --password
  administrative password corresponding to the administrative user.
- --host
  host name of the machine hosting the administrative instance.
- --port
  administrative port number associated with the administrative host.
- --secure
  indicates communication with the administrative instance in secured mode.
- --passwordfile
  file containing passwords appropriate for the command (e.g., administrative instance).
- --instance
  name of the instance.
- --type
  type of service or listener that the SSL is created for. The type can be: http-listener, iiop-listener, and iiop-service.

**OPERANDS**

*listener_id*
the ID of the listener or service that the SSL is created for.

**EXAMPLES**

**EXAMPLE 1** Using delete-ssl

```
asadmin> delete-ssl --user admin --password admin_admin --host fuyako --port 7070 --type http-listener --instance server1 http-listener-1
```

Deleted SSL in HTTP Listener

Where: SSL is deleted for http-listener-1.

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUVALENT**

HTTP Server folder, HTTP Listeners page, ORB folder, IIOP Listeners page

**SEE ALSO**

asadmin-create-ssl(1AS)
NAME
  asadmin-delete-virtual-server, delete-virtual-server – deletes the virtual server with the named virtual server ID

SYNOPSIS
  delete-virtual-server  --user admin_user[--password admin_password]
    [--host localhost] [--port 4848][--passwordfile
      filename] [--secure|-s] [--instance instance_name] virtual_server_ID

DESCRIPTION
  Deletes the virtual server with the named virtual server ID.

OPTIONS
  --user
    administrative user associated for the instance.
  --password
    administrative password corresponding to the administrative user.
  --host
    host name of the machine hosting the administrative instance.
  --port
    administrative port number associated with the administrative host.
  --passwordfile
    file containing passwords appropriate for the command (e.g., administrative instance).
  --secure
    if true, uses SSL/TLS to communicate with the administrative instance.
  --instance
    name of the instance.

OPERANDS
  virtual_server_id
    identifies the unique ID for the virtual server to be created. This virtual server ID cannot begin with a number.

EXAMPLES
  EXAMPLE 1 Using delete-virtual-server

  asadmin> delete-virtual-server --user admin --password adminadmin
  --host localhost --port 4848 --instance server1 sample_vs1
  Deleted virtual server with id = sample_vs1

  Where sample_vs1 is the virtual server deleted.

EXIT STATUS
  0
    command executed successfully
  1
    error in executing the command

INTERFACE
  HTTP Server node, Virtual Servers page

EQUIVALENT
  asadmin-create-virtual-server(1AS), asadmin-list-virtual-servers(1AS)

SEE ALSO
  asadmin-delete-virtual-server(1AS)
asadmin-deploy(1AS)

NAME
asadmin-deploy, deploy – deploys the specified component

SYNOPSIS

DESCRIPTION
Use the deploy command to deploy an EJB, web, connector or application. If the component is already deployed or already exists, it is forcefully re-deployed if the --force option is set to true.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--virtualservers
comma separated list of virtual server IDs.

--type
identifies the type of component to be deployed; defaults to the type of the extension of file.

--contextroot
valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.

--force
makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.

--precompilejsp
by default is set to false which does not allow the JSP to pre-compile during deployment. Instead JSPs are compiled during runtime.

--verify
the syntax and semantics of the deployment descriptor is verified if set to true.

--name
name of the deployable component.
asadmin-deploy(1AS)

**--upload**
when set to true uploads the deployable file to the administration server. If the file path of the deployable file is mounted to the server machine, or if the administration server is running locally, set the upload option to false.

**--retrieve**
retrieves the client stub JAR file from the server machine to the local directory. Retrieve works only if the deployable component is of type application; otherwise it is ignored.

**--instance**
name of the instance.

**filepath**
path to the deployable file on local machine if the **--upload** option is set to true; otherwise the absolute path to the file on the server machine.

**OPERANDS**

**EXAMPLES**

**EXAMPLE 1** Using deploy for WAR module

```
asadmin> deploy --user admin --passwordfile passwords.txt --host localhost
--port 4848 --virtualservers server1 --type web --contextroot simple --no-force
--precompilejsp --verify --name simple --upload --instance server1
/export/samples/simple.war
```

Deployed the WAR module:simple

Where: the simple WAR module is deployed to the server1 instance.

**EXAMPLE 2** Using deploy for an application

```
asadmin> deploy --user admin --password adminadmin --host localhost
--port 4848 --virtualservers server1 --type application --no-force
--precompilejsp --verify --name fortune --upload --instance server1
/export/samples/fortune.ear
```

Deployed the application:fortune

Where: the fortune application is deployed to the absolute file path specified.

**EXIT STATUS**

0 command executed successfully
1 error in executing the command

**INTERFACE EQUIVALENT**
Applications folder, Module interface

**SEE ALSO**

asadmin-deploydir(1AS), asadmin-undeploy(1AS), asadmin-enable(1AS), asadmin-disable(1AS), asadmin-list-components(1)
asadmin-deploydir, deploydir -- deploys the J2EE component that is in the directory located on the server machine

**SYNOPSIS**


**DESCRIPTION**

Use the deploydir command to deploy the J2EE component that is in the directory located on the server machine. The --force option makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists. Set --force to false for a first deployment. If the application with that name is running, and force is set to false, the command fails.

**OPTIONS**

- **--user**
  administrative user associated for the instance.

- **--password**
  administrative password corresponding to the administrative user.

- **--host**
  host name of the machine hosting the administrative instance.

- **--port**
  administrative port number associated with the administrative host.

- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).

- **--secure**
  if true, uses SSL/TLS to communicate with the administrative instance.

- **--virtualservers**
  comma separated list of virtual server IDs.

- **--type**
  identifies the type of component to be deployed; defaults to the type application.

- **--contextroot**
  valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.

- **--force**
  makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.

- **--precompilejsp**
  by default is set to false which does not allow the JSP to pre-compile during deployment. Instead JSPs are compiled during runtime.

- **--verify**
  the syntax and semantics of the deployment descriptor is verified if set to true.
asadmin-deploydir(1AS)

- **--name**
  name of the deployable component.
- **--instance**
  name of the instance.

**OPERANDS**

- **dirpath**
  path to the directory containing the exploded format of the deployable archive.

**EXAMPLES**

**EXAMPLE 1** Using deploydir

```
asadmin> deploydir --user admin --passwordfile passwords.txt --host localhost --port 4848 --force --precompilejsp --verify --name fortune --type application --instance server1 /export/samples/fortune
```

Deployed the application:fortune

Where: the fortune application is deployed to the directory specified.

**EXIT STATUS**

- 0 command executed successfully
- 1 error in executing the command

**INTERFACE EQUIVALENT**

Applications folder, Module interface

**SEE ALSO**

asadmin-deploy(1AS), asadmin-undeploy(1AS), asadmin-enable(1AS), asadmin-disable(1AS), asadmin-list-components(1AS)
asadmin-disable(1AS)

NAME
asadmin-disable, disable – stops the specified component

SYNOPSIS
disable --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type application|ejb|web|connector] [--instance instance_name] component_name

DESCRIPTION
Use the disable command to immediately stop the named component. The component must have been deployed to the specified instance. If the component has not been deployed, an error message is returned.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--type identifies the type of deployed component; defaults to the type application.
--instance name of the instance.

OPERANDS
component_name name of the component to be disabled.

EXAMPLES
EXAMPLE 1 Using disable
asadmin disable --user admin --passwordfile passwords.txt
--host localhost --port 4848 --type web --instance server 1 simple
Disabled the WAR module:simple

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
Applications folder, Module interface

EQUIVALENT
asadmin-deploy(1AS), asadmin-deploydir(1AS), asadmin-undeploy(1AS), asadmin-enable(1AS)

SEE ALSO
Application Server Utility
asadmin-display-license(1AS)

NAME
asadmin-display-license, display-license – displays the license information

SYNOPSIS
display-license [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [-s]

DESCRIPTION
display-license displays the license information. This command can run both locally and remotely.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

EXAMPLES
EXAMPLE 1 Using display-license in local mode
asadmin> display-license
*******************************************************************************
Eval Sun ONE Application Server 7 Evaluation License
Expiration date Tues 11 Sept 11:58:47 PDT 2002
Number of instances per admin server Unlimited
Allow remote administration YES
*******************************************************************************

EXAMPLE 2 Using display-license in remote mode
asadmin> display-license --user admin --password adminadmin
--host fuyako --port 7070
*******************************************************************************
Eval Sun ONE Application Server 7 Evaluation License
Expiration date Tues 11 Sept 11:58:47 PDT 2002
Number of instances per admin server Unlimited
Allow remote administration YES
*******************************************************************************

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-install-license(1AS)
NAME
asadmin-enable, enable – runs the specified component

SYNOPSIS
enable --user admin_user[--password admin_password] [--host localhost]
       [--port 4848] [--secure|-s] [--type application|ejb|web|connector] [--instance instance_name] component_name

DESCRIPTION
Use the enable command to run the specified component. If the component is already enabled, then it is re-enabled. The component must have been deployed in order to be enabled. If it has not been deployed, then an error message is returned.

OPTIONS
--user
   administrative user associated for the instance.
--password
   administrative password corresponding to the administrative user.
--host
   host name of the machine hosting the administrative instance.
--port
   administrative port number associated with the administrative host.
--passwordfile
   file containing passwords appropriate for the command (e.g., administrative instance).
--secure
   if true, uses SSL/TLS to communicate with the administrative instance.
--type
   identifies the type of deployed component; defaults to the type application.
--instance
   name of the instance.

OPERANDS
component_name
   name of the component to be enabled.

EXAMPLES
EXAMPLE 1 Using enable
asadmin enable --user admin --passwordfile passwords.txt
              --host localhost --port 4848 --type web --instance server1 simple
Enabled the WAR module: simple

Where: the simple WAR module is enabled.

EXIT STATUS
0                command executed successfully
1                error in executing the command

INTERFACE
Applications folder, Module interface
EQUIVALENT
asadmin-deploy(1AS), asadmin-deploydir(1AS), asadmin-undeploy(1AS),
asadmin-disable(1AS)

SEE ALSO
Application Server Utility
asadmin-export(1AS)

NAME
asadmin-export, export - marks a variable name for automatic export to the environment of subsequent commands in multimode

SYNOPSIS
export [ name=value [ name=value ]* ]

DESCRIPTION
Use the export command to mark a variable name for automatic export to the environment of subsequent commands. All subsequent commands use the variable name values as specified; unless you unset them or exit multimode. If only the variable name is specified, subsequent commands receive a value set in a previous assignment. If the export command is used without any arguments, a list of all the exported variables and their values is displayed.

Exported shell environment variables set prior to invoking the asadmin utility are imported automatically and set as exported variables within asadmin.

Unexported environment variables cannot be read by the asadmin utility.

OPERANDS
name=value variable name and value for automatic export to the environment to be used by subsequent commands.

EXAMPLES
EXAMPLE 1 Using export to list the environment variables
asadmin> export AS_ADMIN_HOST=bluestar AS_ADMIN_PORT=8000 AS_ADMIN_USER=admin AS_ADMIN_PASSWORD=pass
asadmin> export AS_ADMIN_PREFIX=server1.jms-service
asadmin> export //to list the environment variables that are set
AS_ADMIN_HOST=bluestar
AS_ADMIN_PORT=8000
AS_ADMIN_USER=admin
AS_ADMIN_PASSWORD=********
AS_ADMIN_PREFIX=server1.jms-service

Where: the export command lists the environment variables that are set. In this case, the environment variables have been set to: the host is bluestar, the port is 8000, the administrator user is admin with an associated password, and the prefix is server1.jms-service.

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-unset(1AS), asadmin-multimode(1AS)
Synopsis
```
```

Description
Use the `get` to get the values of attributes. If the `--monitor` option is set to true, the monitorable attributes are returned. If the `--monitor` option is set to false, the configurable attribute values are returned. When using the wildcard character to get multiple attribute values while in single mode, enclose the attribute in double quotes. If you are in multimode, DO NOT use the double quotes.

See the Sun ONE Application Server 7, Administrator’s Guide for a listing of the valid attribute names.

Options
- `--monitor` defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.
- `--user` administrative user associated for the instance.
- `--password` administrative password corresponding to the administrative user.
- `--host` host name of the machine hosting the administrative instance.
- `--port` administrative port number associated with the administrative host.
- `--passwordfile` file containing passwords appropriate for the command (e.g., administrative instance).
- `--secure` if true, uses SSL/TLS to communicate with the administrative instance.

Operands
- `attributename` attribute name in the dotted notation.

Examples
**Example 1** Using `get`
```
asadmin> get --user admin --passwordfile passwords.txt --host localhost
--port 4848 server1.application.fortune.*
server1.application.fortune.location=C:\AS7SE\domains\domain1\server\apps\j2ee-apps\fortune
server1.application.fortune.enabled=true
server1.application.fortune.name=future
server1.application.fortune.description=null
server1.application.fortune.virtualServers=server1
```

Exit Status
- `0` command executed successfully
- `1` error in executing the command

Interface
Anywhere in the Administrator interface
asadmin-get(1AS)

SEE ALSO asadmin-set(1AS), asadmin-reconfig(1AS), asadmin-list(1AS)
asadmin-help, help – displays a list of all the commands available in the Command-line interface

**SYNOPSIS**

`asadmin --help` or `asadmin command --help`

**DESCRIPTION**

Use the help command to display a list of all the commands available in the Command-line interface. Specify the command to display the usage information for that command.

The following is a list of all the Command-line interface commands:

- `multimode`
- `create-instance`
- `delete-instance`
- `export`
- `start-instance`
- `unset`
- `stop-instance`
- `show-instance-status`
- `restart-instance`
- `show-component-status`
- `list-instances`
- `start-appserv`
- `get`
- `stop-appserv`
- `set`
- `list`
- `shutdown`
- `reconfig`
- `help`
- `version`
- `install-license`
- `display-license`
- `add-resources`
- `create-jndi-resource`
- `create-jdbc-connection-pool`
- `delete-jndi-resource`
- `delete-jdbc-connection-pool`
- `list-jndi-resources`
- `list-jdbc-connection-pools`
- `create-iiop-listener`
- `create-jdbc-resource`
- `delete-iiop-listener`
- `delete-jdbc-resource`
- `list-iiop-listeners`
- `list-jdbc-resources`
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create-lifecycle-module</td>
<td>create-persistence-resource</td>
</tr>
<tr>
<td>delete-lifecycle-module</td>
<td>delete-persistence-resource</td>
</tr>
<tr>
<td>list-lifecycle-modules</td>
<td>list-persistence-resources</td>
</tr>
<tr>
<td>create-http-qos</td>
<td>create-mime</td>
</tr>
<tr>
<td>delete-http-qos</td>
<td>delete-mime</td>
</tr>
<tr>
<td>create-virtual-server</td>
<td>list-mimes</td>
</tr>
<tr>
<td>delete-virtual-server</td>
<td>create-authdb</td>
</tr>
<tr>
<td>list-virtual-servers</td>
<td>delete-authdb</td>
</tr>
<tr>
<td>create-ssl</td>
<td>list-authdbs</td>
</tr>
<tr>
<td>delete-ssl</td>
<td>create-auth-realm</td>
</tr>
<tr>
<td>create-domain</td>
<td>delete-auth-realm</td>
</tr>
<tr>
<td>delete-domain</td>
<td>list-auth-realms</td>
</tr>
<tr>
<td>start-domain</td>
<td></td>
</tr>
<tr>
<td>stop-domain</td>
<td>create-jvm-options</td>
</tr>
<tr>
<td>list-domains</td>
<td>delete-jvm-options</td>
</tr>
<tr>
<td>create-jmsdest</td>
<td>deploy</td>
</tr>
<tr>
<td>delete-jmsdest</td>
<td>deploydir</td>
</tr>
<tr>
<td>list-jmsdest</td>
<td>undeploy</td>
</tr>
<tr>
<td>jms-ping</td>
<td>enable</td>
</tr>
<tr>
<td>create-jms-resource</td>
<td>disable</td>
</tr>
<tr>
<td>delete-jms-resource</td>
<td>list-components</td>
</tr>
<tr>
<td>list-jms-resources</td>
<td>list-sub-components</td>
</tr>
<tr>
<td>create-custom-resource</td>
<td>create-javamail-resource</td>
</tr>
<tr>
<td>delete-custom-resource</td>
<td>delete-javamail-resource</td>
</tr>
<tr>
<td>list-custom-resources</td>
<td>list-javamail-resources</td>
</tr>
<tr>
<td>create-acl</td>
<td>create-http-listener</td>
</tr>
</tbody>
</table>
EXAMPLES

EXAMPLE 1 Using the help command

asadmin> help
asadmin> create-instance --help

Where: create-instance is the command you wish to view the usage for.

INTERFACE EQUIVALENT

The Help button on any screen in the Administration interface

SEE ALSO

asadmin-multimode(1AS)
asadmin-install-license(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-install-license, install-license – installs the license file</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>install-license</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>install-license prevents unauthorized use of the Sun ONE Application Server. Allows you to install the license file. This command can be run locally only.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using install-license</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; install-license</td>
</tr>
<tr>
<td></td>
<td>LICENSE agreement will be displayed.</td>
</tr>
<tr>
<td></td>
<td>Do you agree with the terms of this license [YES</td>
</tr>
<tr>
<td></td>
<td>Enter license key&gt; ********</td>
</tr>
<tr>
<td></td>
<td>Installed the license</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-display-license(1AS), asadmin-version(1AS)</td>
</tr>
</tbody>
</table>
asadmin-jms-ping(1AS)

NAME
asadmin-jms-ping, jms-ping -- checks to see if the JMS provider is up and running

SYNOPSIS
jms-ping --user admin_user [--password admin_password] [--host localhost]
          [--port 4848] [--passwordfile filename] [--secure | -s] instance_name

DESCRIPTION
Checks to see if the JMS provider is up and running for the named instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using jms-ping
asadmin> jms-ping --user admin --password adminadmin --host bluestar
          --port 4848 server1
JMS Ping Status=RUNNING

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-create-jmsdest(1AS) asadmin-delete-jmsdest(1AS)
asadmin-list-jmsdest(1AS)
asadmin-list(1AS)

NAME   asadmin-list, list – lists the configurable elements

SYNOPSIS  list[--monitor]--user admin_user[--password admin_password][--host hostname] [--port port_number][--passwordfile filename][--secure|--ssl]element_name

DESCRIPTION Lists the configurable elements (child nodes).

OPTIONS

--monitor              defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.

--user                 administrative user associated for the instance.

--password             administrative password corresponding to the administrative user.

--host                  host name of the machine hosting the administrative instance.

--port                  administrative port number associated with the administrative host.

--passwordfile          file containing passwords appropriate for the command (e.g., administrative instance).

--secure                if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS element_name     configurable or monitorable element name.

EXAMPLES

EXAMPLE 1 Using list for a server instance

asadmin> list --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1
List of configurable attributes for element server1
server1.jndi-resource
server1.persistence-manager-factory-resource
server1.application
server1.http-service
server1.connector-module
server1.transaction-service
server1.iiope-listener
server1.mime
server1.ejb-container
server1.j2ee-application
server1.authrealm
server1.virtual-server-class
server1.acl
server1.mdb-container
server1.external-jndi-resource
server1.http-listener
server1.ORBlistener
server1.java-config
server1.mail-resource
server1.jdbc-resource
EXAMPLE 1 Using list for a server instance (Continued)

server1.iiop-service
server1.jms-service
server1.orb
server1.resources
server1.lifecycle-module
server1.profiler
server1.jms-resource
server1.web-module
server1.custom-resource
server1.virtual-server
server1.jdbc-connection-pool
server1.log-service
server1.security-service
server1.web-container
server1.ejb-module

EXAMPLE 2 Using list for an application

asadmin> list --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1.j2ee-application
List of configurable attributes for element server1.j2ee-application
server1.j2ee-application.fortune

EXAMPLE 3 Using list for a web module

asadmin> list --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1.web-module
List of configurable attributes for element server1.web-module
server1.web-module.simple

EXIT STATUS

0  command executed successfully
1  error in executing the command

INTERFACE EQUIVALENT

Access Control List page

SEE ALSO

asadmin-get(1AS), asadmin-set(1AS), asadmin-reconfig(1AS)
asadmin-list-acls(1AS)

NAME
asadmin-list-acls, list-acls – gets the access control lists for the named instance

SYNOPSIS
list-acls --user admin_user[--password admin_password][--host localhost]
[--port 4848][--passwordfile filename][--secure|-s]instance_name

DESCRIPTION
Gets the access control lists associated with the named server instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--secure
indicates communication with the administrative instance in secured mode.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 Using list-acls
asadmin> list-acls --user admin --password adminadmin
--host fuyako --port 7070 server1
acl1
sampleACL

Where: acl1 and sampleACL are the names of the ACLs listed.

EXIT STATUS
0  command executed successfully
1  error in executing the command

INTERFACE
Access Control List page

EQUIVALENT
SEE ALSO
asadmin-create-acl(1AS), delete-acl(1AS)
asadmin-list-authdbs(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-list-authdbs, list-authdbs – gets the authorized database for the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>list-authdbs --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename] [--secure</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Gets the access control lists associated with the named server instance.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--user administrative user associated for the instance.</td>
</tr>
<tr>
<td></td>
<td>--password administrative password corresponding to the administrative user.</td>
</tr>
<tr>
<td></td>
<td>--host host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--port administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td></td>
<td>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td></td>
<td>--secure if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td></td>
<td>--virtualserver virtual server ID. It can also be referred to as the variable $id in an obj.conf file. A virtual server ID cannot begin with a number.</td>
</tr>
<tr>
<td>OPERANDS</td>
<td>instance_name name of the instance.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using list-authdbs</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; list-authdbs --user admin --password adminadmin --host fuyako --port 7070 --virtualserver server1 server1 default sampleAuth</td>
</tr>
<tr>
<td></td>
<td>Where: default and sampleAuth are the authdb IDs in virtual server server1 and instance server1 listed.</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; list-authdbs --instance server1</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-create-authdb(1AS), asadmin-delete-authdb(1AS)</td>
</tr>
</tbody>
</table>
asadmin-list-auth-realms(1AS)

NAME
asadmin-list-auth-realms, list-auth-realms – lists the authorized realms associated with the named instance

SYNOPSIS
list-auth-realms --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|--s] instance_name

DESCRIPTION
Lists the authorized realms associated with the named instance.

OPTIONS
--user
  administrative user associated for the instance.

--password
  administrative password corresponding to the administrative user.

--host
  host name of the machine hosting the administrative instance.

--port
  administrative port number associated with the administrative host.

--passwordfile
  file containing passwords appropriate for the command (e.g., administrative instance).

--secure
  if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name
  name of the instance.

EXAMPLES
EXAMPLE 1 Using list-auth-realms
asadmin> list-auth-realms --user admin --password admin_password
--host localhost --port 4848 server1
file
ldap
certificate
db

Where file, ldap, certificate, and db are the auth realms listed.

EXIT STATUS
0  command executed successfully
1  error in executing the command

SEE ALSO
asadmin-create-auth-realm(1AS), asadmin-delete-auth-realm(1AS)
asadmin-list-components, list-components – lists deployed J2EE components

SYNOPSIS
list-components --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--type application|ejb|web|connector] instance_name

DESCRIPTION
Use the list-components command to list your deployed J2EE components to the specified instance. If the --type option is not specified, all the components are listed.

OPTIONS
--user
   administrative user associated for the instance.
--password
   administrative password corresponding to the administrative user.
--host
   host name of the machine hosting the administrative instance.
--port
   administrative port number associated with the administrative host.
--passwordfile
   file containing passwords appropriate for the command (e.g., administrative instance).
--secure
   if true, uses SSL/TLS to communicate with the administrative instance.
--type
   identifies the type of component to be listed; defaults to all.

OPERANDS
instance_name
   name of the instance.

EXAMPLES
EXAMPLE 1 Using list-components to list all components
asadmin> list-components --user admin --passwordfile passwords.txt
--port 4848 --host localhost server1
fortune application
simple web
There are no standalone EJB modules
There are no connector modules

Where: all the component that were deployed to the server1 instance are listed.

EXAMPLE 2 Using list-components to list a web component
asadmin> list-components --user admin --passwordfile passwords.txt
--port 4848 --host localhost --type web server1
simple web

Where: all the web component that was deployed to the server1 instance is listed.

EXAMPLE 3 Using list-components to list an application component
asadmin> list-components --user admin --passwordfile passwords.txt
--port 4848 --host localhost --type application server1

EXAMPLE 3 Using `list-components` to list an application component (Continued)

fortune application

Where: all the application component that was deployed to the server1 instance is listed.

**EXIT STATUS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**

Applications folder, Module interface

**SEE ALSO**

`asadmin-list-sub-components(1AS)`, `asadmin-show-component-status(1AS)`
asadmin-list-custom-resources gets all the custom resources from the named instance.

**SYNOPSIS**

```
list-custom-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] instance_name
```

**DESCRIPTION**

Gets all the custom resources from the named instance.

**OPTIONS**

- `--user`   administrative user associated for the instance.
- `--password` administrative password corresponding to the administrative user.
- `--host`   host name of the machine hosting the administrative instance.
- `--port`   administrative port number associated with the administrative host.
- `--passwordfile` file containing passwords appropriate for the command (e.g., administrative instance).
- `--secure` if true, uses SSL/TLS to communicate with the administrative instance.

**OPERANDS**

`instance_name` name of the instance.

**EXAMPLES**

**EXAMPLE 1 Using list-custom-resources**

```
asadmin> list-custom-resources --user admin --password adminadmin
--host fuyako --port 7070 server1
sample_custom_resource
```

Where: `sample_custom_resource` is the custom resource listed.

**EXIT STATUS**

- 0   command executed successfully
- 1   error in executing the command

**INTERFACE EQUIVALENT**

JNDI folder, Custom page

**SEE ALSO**

`asadmin-create-custom-resource(1AS)`, `asadmin-delete-custom-resource(1AS)`
asadmin-list-domains

NAME
asadmin-list-domains, list-domains – lists all the domains

SYNOPSIS
list-domains [--user admin_user] [--password admin_password]
  [--host localhost] [--port 4848] [--local=false]
  [--passwordfile filename] [--secure | -s]

DESCRIPTION
Use the list-domains command to list all the domains associated with the Sun
ONE Application Server.

The list-domains command can be run both locally and remotely. Set the --local
option to true to execute locally. If running remotely, the administrative server must be
running on the hostname specified.

One or more domain must already exist.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the
administrative user.
--host host name of the machine hosting the administrative
instance.
--port administrative port number associated with the
administrative host.
--local determines if the command should delegate the request
to administrative instance or run locally.
--passwordfile file containing passwords appropriate for the command
(e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the
administrative instance.

EXAMPLES
EXAMPLE 1 Using list-domains in local mode
asadmin> list-domains
domain1 [/software/AS7SE/sep9/domains/domain1]
domain2 [/u/mydomain/domain_root/domain2]

Where: the domain1 and domain2 are listed and their directory paths are identified.

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-create-domain(1AS), asadmin-delete-domain(1AS),
asadmin-start-domain(1AS), asadmin-stop-domain(1AS),
asadmin-list-instances(1AS), asadmin-multimode(1AS)
asadmin-list-file-groups(1AS)

NAME
asadmin-list-file-groups, list-file-groups — lists the file groups for the named instance

SYNOPSIS
list-file-groups --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--name username] instance_name

DESCRIPTION
Lists the file groups for the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--name
name of the file user.

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 Using the list-file-groups command
asadmin> list-file-groups --user admin --password adminadmin
--host fuyako --port 7070 --name sample_user server1
staff
manager

Where: staff and manager are the groups for file user sample_user in instance server1.

EXIT STATUS
0
command executed successfully

1
error in executing the command

SEE ALSO
asadmin-delete-file-user(1AS), asadmin-update-file-user(1AS),
asadmin-create-file-user(1AS), asadmin-list-file-users(1AS)
asadmin-list-file-users(1AS)

NAME  asadmin-list-file-users, list-file-users – lists the file users for the named instance

SYNOPSIS  list-file-users --user admin_user [--password admin_password] [--host localhost] [--port port_number] [--passwordfile file_name] [--secure | -s] instance_name

DESCRIPTION  Lists the file users for the named instance.

OPTIONS
   --user  administrative user associated for the instance.
   --password  administrative password corresponding to the administrative user.
   --host  host name of the machine hosting the administrative instance.
   --port  administrative port number associated with the administrative host.
   --passwordfile  file containing passwords appropriate for the command (e.g., administrative instance).
   --secure  if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS  instance_name  name of the instance.

EXAMPLES  EXAMPLE 1 Using the list-file-users command
   asadmin> list-file-users --user admin --password adminadmin --host fuyako --port 7070 server1 sample_user
   Where: the sample_user is the file user listed.

EXIT STATUS  0  command executed successfully
   1  error in executing the command

SEE ALSO  asadmin-delete-file-user(1AS), asadmin-update-file-user(1AS), asadmin-create-file-user(1AS), asadmin-list-file-groups(1AS)
asadmin-list-http-listeners(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-list-http-listeners, list-http-listeners – gets the HTTP listeners for the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td><code>list-http-listeners --user admin_user[--password admin_password][--host localhost][--port 4848][--passwordfile filename][--secure][--instance instance_name] http_listener_ID</code></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Gets the HTTP listeners associated with the named server instance.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td><code>-user</code> administrative user associated for the instance.</td>
</tr>
<tr>
<td></td>
<td><code>-password</code> administrative password corresponding to the administrative user.</td>
</tr>
<tr>
<td></td>
<td><code>-host</code> host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td></td>
<td><code>-port</code> administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td></td>
<td><code>-passwordfile</code> file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td></td>
<td><code>-secure</code> if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td></td>
<td><code>-instance</code> name of the instance.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td><strong>EXAMPLE 1</strong> Using list-http-listeners</td>
</tr>
<tr>
<td></td>
<td><code>asadmin&gt; list-http-listeners --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sampleListener</code></td>
</tr>
<tr>
<td></td>
<td>Deleted HTTP listener with id = sampleListener</td>
</tr>
<tr>
<td></td>
<td>Where: sampleListener is the HTTP listener listed.</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>INTERFACE EQUIVALENT</td>
<td>HTTP Server folder, HTTP Listener page</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-create-http-listener(1AS), asadmin-delete-http-listener(1AS)</td>
</tr>
</tbody>
</table>
asadmin-list-iiop-listeners(1AS)

NAME
asadmin-list-iiop-listeners, list-iiop-listeners – gets the IIOP listeners for the named instance

SYNOPSIS
list-iiop-listeners --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename][--secure|--s] instance_name

DESCRIPTION
Gets the IIOP listeners associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using list-iiop-listeners
asadmin> list-iiop-listeners --user admin --password adminadmin
--host fuyako --port 7070 server1
orb-listener-1
sample_iiop_listener

Where: orb-listener-1 and sample_iiop_listener are the IIOP listeners listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
ORB folder, IIOP Listener page

EQUIVALENT
SEE ALSO
asadmin-create-iiop-listener(1AS),
asadmin-delete-iiop-listener(1AS)
asadmin-list-instances(1AS)

NAME
asadmin-list-instances, list-instances – lists all the instances in the server

SYNOPSIS
list-instances [--user admin_user] [--password admin_password]
               [--host localhost] [--port 4848] [--domain domain_name]
               [--local=false] [--passwordfile filename] [--secure] -s

DESCRIPTION
Use the list-instances to list all the instance in the server. The list-instances command can be run both locally and remotely.

To list remote instances, the named administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server.

OPTIONS
--user       administrative user associated for the instance.
--password   administrative password corresponding to the administrative user.
--host       host name of the machine hosting the administrative instance.
--port       port number associated with the administrative host.
--domain     name of the domain.
--local      determines if the command should delegate the request to administrative instance or run locally.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure     if true, uses SSL/TLS to communicate with the administrative instance.

EXAMPLES
EXAMPLE 1 Using list-instances in local mode
asadmin> list-instances --domain1 --local
admin-server running
server1 running

Where: the server1 and admin-server instances for the domain1 domain is listed.

EXAMPLE 2 Using list-instances in remote mode
asadmin> list-instances --user admin --passwordfile passwords.txt --host localhost --port 4848
server1 [mayank:80] running

Where: the server1 instance associated with the specified user, passwords, host, and port number specified is listed for the remote machine.

EXIT STATUS
0 command executed successfully
1 error in executing the command
asadmin-list-instances(1AS)

INTERFACE EQUIVALENT
Server Instance page
SEE ALSO
asadmin-show-instance-status(1AS)
asadmin-list-javamail-resources(1AS)

**NAME**
asadmin-list-javamail-resources, list-javamail-resources – gets all the Javamail resources from the named instance

**SYNOPSIS**

```
list-javamail-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name
```

**DESCRIPTION**

Gets all the Javamail resources from the named instance.

**OPTIONS**

- **--user**
  administrative user associated for the instance.

- **--password**
  administrative password corresponding to the administrative user.

- **--host**
  host name of the machine hosting the administrative instance.

- **--port**
  administrative port number associated with the administrative host.

- **--passwordfile**
  file containing passwords appropriate for the command (e.g., administrative instance).

- **--secure**
  if true, uses SSL/TLS to communicate with the administrative instance.

**OPERANDS**

```
instance_name
```

name of the instance.

**EXAMPLES**

**EXAMPLE 1 Using list-javamail-resources**

```
asadmin> list-javamail-resources --user admin --password adminadmin
--host fuyako --port 7070 server1
sample_javamail_resource
```

Where: **sample_javamail_resource** is the Javamail resource listed.

**EXIT STATUS**

- 0 command executed successfully
- 1 error in executing the command

**INTERFACE EQUIVALENT**

Javamail Sessions page

**SEE ALSO**

asadmin-create-javamail-resource(1AS)
asadmin-delete-javamail-resource(1AS)
asadmin-list-jdbc-connection-pools(1AS)

NAME
asadmin-list-jdbc-connection-pools, list-jdbc-connection-pools – gets all the JDBC connection pools from the named instance

SYNOPSIS
list-jdbc-connection-pools --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] instance_name

DESCRIPTION
Gets all the JDBC resources connection pools from the named instance.

OPTIONS
- --user
  administrative user associated for the instance.
- --password
  administrative password corresponding to the administrative user.
- --host
  host name of the machine hosting the administrative instance.
- --port
  administrative port number associated with the administrative host.
- --passwordfile
  file containing passwords appropriate for the command (e.g., administrative instance).
- --secure
  if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 Using list-jdbc-connection-pools
asadmin> list-jdbc-connection-pools --user admin --password adminadmin
--host fuyako --port 7070 server1
XA_connection_pool

Where: XA_connection_pool is the JDBC connection listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
JDBC folder, Connection Pool page

EQUIVALENT

SEE ALSO
asadmin-create-jdbc-connection-pool(1AS),
asadmin-delete-jdbc-connection-pool(1AS)
asadmin-list-jdbc-resources(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-list-jdbc-resources, list-jdbc-resources – gets all the JDBC resources from the named instance</th>
</tr>
</thead>
</table>
| SYNOPTIS | **list-jdbc-resources**  
**--user** *admin_user*  
**--password** *admin_password*  
**--host** *localhost*  
**--port** 4848  
**--passwordfile** *filename*  
**--secure**  
**instance_name** |
| DESCRIPTION | Gets all the JDBC resources from the named instance. |
| OPTIONS |  
**--user**  
administrative user associated for the instance. |
|  
**--password**  
administrative password corresponding to the administrative user. |
|  
**--host**  
host name of the machine hosting the administrative instance. |
|  
**--port**  
administrative port number associated with the administrative host. |
|  
**--passwordfile**  
file containing passwords appropriate for the command (e.g., administrative instance). |
|  
**--secure**  
if true, uses SSL/TLS to communicate with the administrative instance. |
| OPERANDS | **instance_name**  
name of the instance. |
| EXAMPLES | **EXAMPLE 1** Using the list-jdbc-resources command  
asadmin> list-jdbc-resources --user --password adminadmin  
--host fuyako --port 7070 server1  
sample_jdbc_resource  
Where: sample_jdbc_resource is the JDBC connection listed. |
| EXIT STATUS |  
0  
command executed successfully  
1  
error in executing the command |
| INTERFACE EQUIVALENT | JDBC folder, Datasource page |
| SEE ALSO | asadmin-create-jdbc-resource(1AS), asadmin-delete-jdbc-resource(1AS) |
asadmin-list-jmsdest(1AS)

NAME
asadmin-list-jmsdest, list-jmsdest – gets all the named destinations

SYNOPSIS
list-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--desttype type] instance_name

DESCRIPTION
Gets all the named destinations.

OPTIONS
- --user administrative user associated for the instance.
- --password administrative password corresponding to the administrative user.
- --host host name of the machine hosting the administrative instance.
- --port administrative port number associated with the administrative host.
- --passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
- --secure if true, uses SSL/TLS to communicate with the administrative instance.
- desttype type of JMS destination. Valid values are topic and queue.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using list-jmsdest
asadmin> list-jmsdest --user admin --password adminpass --host bluestar --port 4848 server1

exit_status
0 command executed successfully
1 error in executing the command

INTERFACE
JMS Destination pages

EQUIVALENT

SEE ALSO
asadmin-create-jmsdest(1AS), asadmin-delete-jmsdest(1AS)
asadmin-list-jms-resources(1AS)

NAME
asadmin-list-jms-resources, list-jms-resources – gets all the JMS resources from the named instance

SYNOPSIS
list-jms-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [--resourcetype type] instance_name

DESCRIPTION
Gets all the JMS resources from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--resourcetype

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 Using the list-jms-resources command
asadmin> list-jms-resources --user admin --password adminadmin
--host fuyako --port 7070 --resourcetype javax.jms.Queue server1
sample_jms_resource

Where: sample_jms_resource is the JMS resource listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
JMS folder, Destinations page

SEE ALSO
asadmin-create-jms-resource(1AS), asadmin-delete-jms-resource(1AS)
asadmin-list-jndi-resources(1AS)

NAME
asadmin-list-jndi-resources, list-jndi-resources – gets all the JNDI resources from the named instance

SYNOPSIS
list-jndi-resources [options] instance_name

DESCRIPTION
Gets all the JNDI resources from the named instance.

OPTIONS
- --user
  administrative user associated for the instance.
- --password
  administrative password corresponding to the administrative user.
- --host
  host name of the machine hosting the administrative instance.
- --port
  administrative port number associated with the administrative host.
- --passwordfile
  file containing passwords appropriate for the command (e.g., administrative instance).
- --secure
  if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPE 1 Using the list-jndi-resource command
asadmin> list-jndi-resource --user admin --password adminadmin
--host fuyako --port 7070 server1
sample_jndi_resource

Where: sample_jndi_resource is the JNDI resource listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
JNDI folder, External page

SEE ALSO
asadmin-create-jndi-resource(1AS),
asadmin-delete-jndi-resource(1AS)
asadmin-list-lifecycle-modules

NAME
asadmin-list-lifecycle-modules, list-lifecycle-modules – gets the lifecycle modules for the named instance.

SYNOPSIS
list-lifecycle-modules --user admin_user[--password admin_password]
    [--host localhost] [--port 4848][--passwordfile filename][--secure|-s][--instance instance_name] module_name

DESCRIPTION
Gets the lifecycle modules associated with the named server instance.

OPTIONS
--user
    administrative user associated for the instance.
--password
    administrative password corresponding to the administrative user.
--host
    host name of the machine hosting the administrative instance.
--port
    administrative port number associated with the administrative host.
--passwordfile
    file containing passwords appropriate for the command (e.g., administrative instance).
--secure
    if true, uses SSL/TLS to communicate with the administrative instance.
instance_name
    name of the instance.

EXAMPLES
EXAMPLE 1 Using list-lifecycle-modules
asadmin> list-lifecycle-modules --user admin --password adminadmin
    --host fuyako --port 7070 server1
    customSetup
Where: customSetup is the lifecycle module listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Application Lifecycle Modules page

SEE ALSO
asadmin-create-lifecycle-module(1AS),
asadmin-delete-lifecycle-module(1AS)
asadmin-list-mimes(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-list-mimes, list-mimes -- gets the MIME types for the named instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>list-mimes --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] instance_name</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Gets the MIME types associated with the named server instance. The server determines the MIME type of a requested resource by invoking the type-by-extension directive in the ObjectType section of the obj.conf file. The type-by-extension function does not work if no MIME element has been defined in the server element.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--user administrative user associated for the instance. --password administrative password corresponding to the administrative user. --host host name of the machine hosting the administrative instance. --port administrative port number associated with the administrative host. --passwordfile file containing passwords appropriate for the command (e.g., administrative instance). --secure if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
<tr>
<td>OPERANDS</td>
<td>instance_name name of the instance.</td>
</tr>
</tbody>
</table>
| EXAMPLES | EXAMPLE 1 Using list-mimes
asadmin> list-mimes --user admin --password adminadmin --host fuyako --port 7070 server1 sampleMIME
Where: sampleMIME is the name of the MIME listed. |
| EXIT STATUS | 0 command executed successfully 1 error in executing the command |
| INTERFACE | HTTP Server node, MIME Type Files page |
| EQUIVALENT | |
| SEE ALSO | asadmin-create-mime(1AS), asadmin-delete-mime(1AS) |

144 Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
asadmin-list-persistence-resources(1AS)

NAME
asadmin-list-persistence-resources, list-persistence-resources – gets all the persistence resources from the named instance

SYNOPSIS
list-persistence-resources --user admin_user[--password admin_password] [--host localhost][--port 4848][--passwordfile filename][--secure|-s] instance_name

DESCRIPTION
Gets all the persistence resources from the named instance.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 using list-persistence-resources

asadmin> list-persistence-resources --user admin --password adminadmin
--host fuyako --port 7070 server1
sample_persistence_resource

Where: sample_persistence_resource is the persistence manager factory resource listed.

EXIT STATUS
0  command executed successfully
1  error in executing the command

INTERFACE
Persistence Manager page

EQUIVALENT

SEE ALSO
asadmin-create-persistence-resource(1AS),
asadmin-delete-persistence-resource(1AS)
asadmin-list-profilers(1AS)

NAME
asadmin-list-profilers, list-profilers – lists the profiler elements in the named instance

SYNOPSIS
list-profilers --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure] [instance_name]

DESCRIPTION
Gets the profiler element associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using list-profilers
asadmin> list-profilers --user admin --passwordfile passwords.txt --host localhost --port 4848 server1 sample_profiler

Where: sample_profiler is the profiler listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
Application Server Instances, JVM Settings tab

EQUIVALENT
SEE ALSO
asadmin-create-profiler(1AS) asadmin-delete-profiler(1AS)
asadmin-list-sub-components(1AS)

NAME
asadmin-list-sub-components, list-sub-components -- lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application.

SYNOPSIS

DESCRIPTION
Use the list-sub-components to list your EJBs or Servlets in a deployed module or in a module of the deployed application. If a module is not identified, all modules are listed. The component type defaults to EJBs.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--type identifies the type of sub-component to be listed.
--instance name of the instance.
--appname name of the application.

OPERANDS
module_name name of the module containing the sub-components.

EXAMPLES
EXAMPLE 1 Using list-sub-components
asadmin list-sub-components --user admin --passwordfile passwords.txt
--port 4848 --host localhost --instance server1 --type servlets --appname fortune fortune

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Applications folder, Module interface

SEE ALSO
asadmin-deploy(1AS), asadmin-deploydir(1AS), asadmin-undeploy(1AS), asadmin-enable(1AS), asadmin-disable(1AS), asadmin-list-components(1AS)
NAME
asadmin-list-virtual-servers, list-virtual-servers – gets the virtual servers in the named instance

SYNOPSIS
list-virtual-servers --user admin_user[--password admin_password][--host localhost] [--port 4848][--passwordfile filename][--secure|--s] instance_name

DESCRIPTION
Gets the virtual server elements associated with the named server instance.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance.

EXAMPLES
EXAMPLE 1 Using list-virtual-servers
asadmin> list-virtual-servers --user admin --password adminadmin
--host localhost --port 4848 server1
server1
sample_vs1

Where server1 and sample_vs1 are the virtual servers listed.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
HTTP Server node, Virtual Servers page

SEE ALSO
asadmin-create-virtual-server(1AS),
asadmin-delete-virtual-server(1AS)
asadmin-multimode — allows you to execute multiple commands while returning environment settings and remaining in the asadmin utility

SYNOPSIS

multimode [--file filename] [--encoding encode]
    [--passwordfile filename] [--interactive]

DESCRIPTION

Use the multimode command to set your Command-line interface environment settings so you can run multiple commands without having to re-enter the environment level information. In multimode, you can set your environment and run commands until you exit multimode by typing “exit” or “quit”. Additionally, you can provide commands by passing a previously prepared list of commands from a file or standard input (pipe). You can invoke multimode from within a multimode session; once you exit the second multimode environment, you return to your original multimode environment.

The interactive (default) option prompts you for the required arguments. Use the interactive option when you run one command at a time from the command prompt or when you run in multimode from a file. Commands in multimode, when piped from an input stream, and commands invoked from another program, cannot run in the interactive mode.

OPTIONS

--file
    consists of commands to be executed in multimode.

--encoding
    system locale encoding method to be used.

--passwordfile
    file containing the administrative passwords appropriate for the command (e.g., administrative instance)

--interactive
    prompts you for the required options.

EXAMPLES

EXAMPLE 1 Using multimode to execute multiple commands

example% asadmin multimode --file commands_file.txt

Where: example% is the system prompt. The multimode settings are executed from the commands_file.txt file.

EXIT STATUS

0
    command executed successfully

1
    error in executing the command

SEE ALSO

asadmin-export(1AS), asadmin-unset(1AS)
asadmin-reconfig(1AS)

NAME
asadmin-reconfig, reconfig -- applies the changes you have made for a server instance

SYNOPSIS

DESCRIPTION
reconfig allows you to apply changes you have made for a server instance. Use the reconfig command after you’ve used the set command to change server properties. Any changes you make to the configuration files of the server do not take affect until you apply the changes by running the reconfig command.

When --discardmanualchanges is set to true, manual changes made to the server.xml file are discarded. When --keepmanualchanges is set to true, manual changes made to the server.xml file take affect. However if both options are false (both options are not specified), an error message is displayed if manual changes and/or changes have been applied using the Administrator Interface.

Use this command with discretion since there is no undo, and the changes applied are made directly to your config/backup/server.xml file.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--discardmanualchanges
defaults to false. When set to true, discards the changes made manually to the server.xml file.

--keepmanualchanges
defaults to false. When set to true, allows the manual changes made to the server.xml file to take affect.

OPERANDS
instance_name
name of the instance.

EXAMPLES
EXAMPLE 1 Using reconfig
asadmin> reconfig --user admin --passwordfile passwords.txt
--host localhost --port 4848 server1
Successfully reconfigured

150 Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
EXAMPLE 2 Using `reconfig` with the `--discardmanualchanges` option

```
asadmin> reconfig --user admin --passwordfile passwords.txt
--host localhost --port 4848 --discardmanualchanges server1
Instance restart is required
Successfully reconfigured
```

EXAMPLE 3 Using `reconfig` with the `--keepmanualchanges` option

```
asadmin> reconfig --user admin --passwordfile passwords.txt
--host localhost --port 4848 --keepmanualchanges server1
Instance restart is required
Successfully reconfigured
```

EXIT STATUS

- 0: Command executed successfully
- 1: Error in executing the command

INTERFACE EQUIVALENT

Any Apply Changes button in the interface

SEE ALSO

`asadmin-get(1AS) asadmin-set(1AS) asadmin-list(1AS)`
asadmin-restart-instance, restart-instance

DESCRIPTION
Use the restart-instance to restart the instance with the instance name specified. The restart-instance command can be run both locally and remotely.

To restart remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server, and the instance must be running.

The restart-instance command is not supported on Windows.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--local determines if the command should delegate the request to administrative instance or run locally.
--domain name of the domain.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance to be restarted.

EXAMPLES
EXAMPLE 1 Using restart-instance in local mode
asadmin> restart-instance --local --domain domain1 server1
Instance server1 started

Where: server1 is the name of the instance restarted on the domain1 domain.

EXAMPLE 2 Using restart-instance in remote mode
asadmin> restart-instance --user admin --password adminadmin --host bluestar --port 4848 server1
Instance server1 started
EXAMPLE 2 Using restart-instance in remote mode (Continued)

Where: server1 is the name of the instance restarted. The restarted instance is associated with the user, password, host, and port number specified.

<table>
<thead>
<tr>
<th>EXIT STATUS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

Server Instance page

asadmin-delete-instance(1AS), asadmin-start-instance(1AS), asadmin-create-instance(1AS), asadmin-stop-instance(1AS), asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS)
asadmin-set(1AS)

**NAME**
asadmin-set, set – sets the values of attributes

**SYNOPSIS**

**DESCRIPTION**
Sets the values of of one or more configurable attribute. The settings do not take affect until you run the reconfig command.

**OPTIONS**
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--monitor</td>
<td>defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.</td>
</tr>
<tr>
<td>--user</td>
<td>administrative user associated for the instance.</td>
</tr>
<tr>
<td>--password</td>
<td>administrative password corresponding to the administrative user.</td>
</tr>
<tr>
<td>--host</td>
<td>host name of the machine hosting the administrative instance.</td>
</tr>
<tr>
<td>--port</td>
<td>administrative port number associated with the administrative host.</td>
</tr>
<tr>
<td>--passwordfile</td>
<td>file containing passwords appropriate for the command (e.g., administrative instance).</td>
</tr>
<tr>
<td>--secure</td>
<td>if true, uses SSL/TLS to communicate with the administrative instance.</td>
</tr>
</tbody>
</table>

**OPERANDS**
attributename=value identifies the configurable or monitorable attribute name and its value.

**EXAMPLES**
**EXAMPLE 1 Using set**
```
asadmin> set --user admin --passwordfile passwords.txt --host localhost --port 4848 server1.application.fortune.enabled=false
Attribute enabled set to false
```

**EXIT STATUS**
<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

**INTERFACE EQUIVALENT**
Anywhere in the interface

**SEE ALSO**
asadmin-get(1AS), asadmin-reconfig(1AS), asadmin-list(1AS)
asadmin-show-component-status(1AS)

NAME

asadmin-show-component-status, show-component-status – displays the status of the deployed component

SYNOPSIS

```
show-component-status --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name] component_name
```

DESCRIPTION

Use the `show-component-status` command to get the status of the deployed component. The status is a string representation returned by the server. The possible status strings include: enabled or disabled.

OPTIONS

```
--user
    administrative user associated for the instance.
--password
    administrative password corresponding to the administrative user.
--host
    host name of the machine hosting the administrative instance.
--port
    administrative port number associated with the administrative host.
--passwordfile
    file containing passwords appropriate for the command (e.g., administrative instance).
--secure
    if true, uses SSL/TLS to communicate with the administrative instance.
--instance
    name of the instance.
```

OPERANDS

```
component_name
    name of the component to be listed.
```

EXAMPLES

**EXAMPLE 1** Using `show-component-status` to show an application

```
asadmin> show-component-status --user admin --passwordfile passwords.txt --host bluestar --port 4848 fortune
Status of application fortune is enabled
```

Where: the status of the `fortune` application is shown.

**EXAMPLE 2** Using `show-component-status` to show a WAR module

```
asadmin> show-component-status --user admin --passwordfile passwords.txt --host bluestar --port 4848 simple
Status of WAR module simple is enabled
```

Where: the status of the `simple` WAR module is shown.

EXIT STATUS

```
0
    command executed successfully
1
    error in executing the command
```

INTERFACE

```
Web, EJB, Connector, and Lifecycle modules
```

Application Server Utility 155
See Also

asadmin-list-components(1AS), asadmin-list-sub-components(1AS)
**NAME**  
asadmin-show-instance-status, show-instance-status – displays the status of the server instance specified

**SYNOPSIS**  
```
show-instance-status --user admin_user[--password admin_password] [--host localhost] [--port 4848][--local=false][--passwordfile filename] [--secure|-s] instance_name
```

**DESCRIPTION**  
Use the show-instance-status command to get the status of the specified instance. The instance must already exist. If the instance specified does not exist, the command fails. The status is a string representation returned by the server; it can be: starting, started, stopping, and stopped.

**OPTIONS**  
- `--user` administrative user associated for the instance.
- `--password` administrative password corresponding to the administrative user.
- `--host` host name of the machine hosting the administrative instance.
- `--port` administrative port number associated with the administrative host.
- `--passwordfile` file containing passwords appropriate for the command (e.g., administrative instance).
- `--secure` if true, uses SSL/TLS to communicate with the administrative instance.

**OPERANDS**  
`instance_name` name of the instance.

**EXAMPLES**  
**EXAMPLE 1** Using show-instance-status
```
asadmin> show-instance-status --user admin --password adminadmin
--host localhost --port 4848 server1
```
Status of instance server1 is running

Where: the status of the server1 instance is shown.

**EXIT STATUS**  
0 command executed successfully
1 error in executing the command

**INTERFACE EQUIVALENT**  
Server Instance page; Applications folder, Enterprise applications

**SEE ALSO**  
asadmin-list-instances(1AS)
asadmin-shutdown(1AS)

NAME
asadmin-shutdown, shutdown – brings down the administration server and associated instances

SYNOPSIS
shutdown [-user admin_user] [-password admin_password] [-host localhost] [-port 4848] [-passwordfile filename] [-secure|-s]

DESCRIPTION
shutdown gracefully brings down the administration server and all the running instances. You must manually start the administration server to bring it up again.

OPTIONS
- --user administrative user associated for the instance.
- --password administrative password corresponding to the administrative user.
- --host host name of the machine hosting the administrative instance.
- --port administrative port number associated with the administrative host.
- --passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
- --secure if true, uses SSL/TLS to communicate with the administrative instance.

EXAMPLES
EXAMPLE 1 Using the shutdown command
asadmin> shutdown --user admin --password admin_admin --host bluestar --port 4848
Waiting for admin server to shutdown...
Admin server has been shutdown

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE EQUIVALENT
Administration Server page

SEE ALSO
asadmin-start-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS), asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS)
asadmin-start-appserv

NAME
asadmin-start-appserv, start-appserv – starts the local Administration server and all the instances associated with it

SYNOPSIS
start-appserv

DESCRIPTION
Use the start-appserv command to start all the domains defined for the application server installation; use with caution. The start-appserv command can be run locally only. One or more domain must already exist.

EXAMPLES
EXAMPLE 1 Using start-appserv

    asadmin> start-appserv
    Instance domain1:admin-server started
    Instance domain1:server1 started
    Domain domain1 started
    Instance sample_domain:admin-server started
    Domain sample_domain started

    Where: the admin-server and server1 instances are started along with the domain domain1 they are associated with. The admin-server instance and the sample-domain domain it is associated with are also started.

EXIT STATUS
0    command executed successfully
1    error in executing the command

SEE ALSO
asadmin-create-instance(1AS), asadmin-delete-instance(1AS), asadmin-start-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS), asadmin-stop-appserv(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS)
asadmin-start-domain(1AS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>asadmin-start-domain, start-domain – starts the given domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td>start-domain [--domain domain_name]</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Use the start-domain command to start all the instances in the specified domain. If the --domain option is not specified, and there is only one domain, all the instances in that domain are started. The start-domain command can be run locally only. The domain must currently exist on the local machine.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>--domain name of the domain; must be a unique name.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>EXAMPLE 1 Using start-domain</td>
</tr>
<tr>
<td></td>
<td>asadmin&gt; start-domain --domain domain1</td>
</tr>
<tr>
<td></td>
<td>instance domain1:admin-server started</td>
</tr>
<tr>
<td></td>
<td>instance domain1:server1 started</td>
</tr>
<tr>
<td></td>
<td>domain domain1 started</td>
</tr>
<tr>
<td></td>
<td>Where: the domain1 domain is started. By starting the domain, the admin-server and server1 instances in the domain are also started.</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>0 command executed successfully</td>
</tr>
<tr>
<td></td>
<td>1 error in executing the command</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>asadmin-create-domain(1AS), asadmin-delete-domain(1AS),</td>
</tr>
<tr>
<td></td>
<td>asadmin-stop-domain(1AS), asadmin-list-domains(1AS),</td>
</tr>
<tr>
<td></td>
<td>asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS),</td>
</tr>
<tr>
<td></td>
<td>asadmin-start-instance(1AS), asadmin-stop-instance(1AS)</td>
</tr>
</tbody>
</table>
NAME
asadmin-start-instance, start-instance – starts a server instance and all the services associated with it

SYNOPSIS
start-instance [--user admin_user] [--password admin_password]
              [--host local_host] [--port 4848] [--local=false]
              [--domain domain_name] [--debug=false] [--passwordfile filename]
              [--secure|-s] instance_name

DESCRIPTION
Use the start-instance command to start an instance with the instance name you specify. The start-instance command can be run both locally and remotely. To start locally, with a domain name identified, the named instance must already exist within that domain. To start remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--local determines if the command should delegate the request to administrative instance or run locally.
--domain name of the domain.
--debug starts the instance in debug mode.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance to be started.

EXAMPLES
EXAMPLE 1 Using start-instance in local mode
asadmin> start-instance --domain domain1 admin-server
Instance admin-server started
Where: the admin-server instance is started on the local domain1 domain.

EXAMPLE 2 Using start-instance in remote mode
asadmin> start-instance --user admin --password bluestar --host localhost --port 4848 server1
Instance server1 started
EXAMPLE 2 Using start-instance in remote mode (Continued)

Where: the server1 instance is started on the remote domain associated with the specified user, password, host, and port number.

<table>
<thead>
<tr>
<th>EXIT STATUS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

INTERFACE EQUIVALENT

SEE ALSO

- asadmin-delete-instance(1AS), asadmin-create-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS), asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS)
asadmin-stop-appserv

NAME
asadmin-stop-appserv, stop-appserv – stops the local administration server and all the instances associated with it

SYNOPSIS
stop-appserv

DESCRIPTION
Use the stop-appserv command to stop all the domains, and its instances, in the application server installation; use with caution. The stop-appserv can be run locally only. One or more domain must already exist.

EXAMPLES
EXAMPLE 1 Using stop-appserv

asadmin> stop-appserv
Instance domain1:admin-server stopped
Instance domain1:server1 stopped
Domain domain1 stopped
Instance sample_domain:admin-server stopped
Domain sample_domain stopped

Where: the admin-server and server1 instances are stopped along with the domain domain1 they are associated with. The admin-server instance and the sample-domain domain it is associated with are also stopped.

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-create-instance(1AS), asadmin-delete-instance(1AS), asadmin-start-instance(1AS), asadmin-stop-instance(1AS), asadmin-restart-instance(1AS), asadmin-start-appserv(1AS), asadmin-start-domain(1AS), asadmin-stop-domain(1AS)
asadmin-stop-domain(1AS)

NAME
asadmin-stop-domain, stop-domain – stops the given domain

SYNOPSIS
stop-domain [--user admin_user] [--password admin_password]
            [--host local_host] [--port 4848] [--local=false]
            [--domain domain_name] [--adminserv=true]
            [--passwordfile filename] [--secure|-s]

DESCRIPTION
Use the stop-domain command to stop all the instances in the domain specified.

The stop-domain command can be run both locally and remotely. The domain must exist on the local machine to run this command locally.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--local determines if the command should delegate the request to administrative instance or run locally.
--domain name of the domain; must be a unique name. If not specified, and there is only one domain, all instances in that domain are stopped.
--adminserv determines if the administrative instance should be stopped along with other instances.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if set to true, uses SSL/TLS to communicate with the administrative instance.

EXAMPLES
EXAMPLE 1 Using stop-domain in local mode
asadmin> stop-domain --domain domain1 --adminserv=true
Instance domain1:admin-server stopped
Instance domain1:server1 stopped
Domain domain1 stopped

Where: the domain1 domain is stopped. By stopping the domain the admin-server and server1 instance in the domain are also stopped.

EXAMPLE 2 Using stop-domain in remote mode
asadmin> stop-domain --user admin --passwordfile passwords.txt --host bluestar --port 6886
Domain stopped remotely
EXAMPLE 2 Using stop-domain in remote mode  (Continued)

Where: the domain identified with the user, host, and port specified is stopped on the remote server. All instances in the domain are also stopped.

EXIT STATUS

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>command executed successfully</td>
</tr>
<tr>
<td>1</td>
<td>error in executing the command</td>
</tr>
</tbody>
</table>

SEE ALSO

asadmin-create-domain(1AS), asadmin-delete-domain(1AS),
asadmin-start-domain(1AS), asadmin-list-domains(1AS),
asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS),
asadmin-start-instance(1AS), asadmin-stop-instance(1AS),
asadmin-multimode(1AS)
asadmin-stop-instance(1AS)

NAME
asadmin-stop-instance, stop-instance – stops the specified server instance and all the services associated with it.

SYNOPSIS
stop-instance [--user admin_user] [--password admin_password]
[--host local_host] [--port 4848] [--local=false]
[--domain domain_name] [--secure|-s] instance_name

DESCRIPTION
Use the stop-instance to stop the instance with the instance name specified. The stop-instance can be run both locally and remotely.

The named instance must already exist within the given domain; and the instance must be running.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--local determines if the command should delegate the request to administrative instance or run locally.
--domain name of the domain.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

OPERANDS
instance_name name of the instance to be stopped.

EXAMPLES
EXAMPLE 1 Using stop-instance in local mode
asadmin> stop-instance --local --domain domain1 server1
Instance server1 stopped

Where: the server1 instance associated with the domain1 domain is stopped locally.

EXAMPLE 2 Using stop-instance in remote mode
asadmin> stop-instance --user admin --password bluestar --host localhost --port 4848 server1
Instance server1 stopped

Where: the server1 instance associated with the named user, password, host and port is deleted from the remote machine.

EXIT STATUS
0 command executed successfully

Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
error in executing the command

Server Instance page

asadmin-delete-instance(1AS), asadmin-start-instance(1AS),
asadmin-create-instance(1AS), asadmin-restart-instance(1AS),
asadmin-start-appserv(1AS), asadmin-stop-appserv(1AS),
asadmin-start-domain(1AS), asadmin-stop-domain(1AS)
asadmin-undeploy(1AS)

NAME
asadmin-undeploy, undeploy – removes the component from the named instance.

SYNOPSIS
undeploy --user admin_user[--password admin_password][--host localhost]
[ --port 4848][--passwordfile filename][--secure|-s][--type application|ejb|web|connector][--instance instance_name]
component_name

DESCRIPTION
Use the undeploy command to remove the specified component. You can specify the component type that you wish to remove and the instance that the component was deployed to.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.
--type identifies the type of component to be deployed; defaults to the type application.
--instance name of the instance.

OPERANDS
component_name name of the deployable component.

EXAMPLES
EXAMPLE 1 Using undeploy
asadmin> undeploy --user admin --password adminadmin --host localhost
--port 4848 --type application --instance server1 fortune
Undeployed the application:fortune

Where: the fortune application is undeployed from the server1 instance.

EXIT STATUS
0 command executed successfully
1 error in executing the command

INTERFACE
Applications folder, Module interface

EQUIVALENT
SEE ALSO
asadmin-deploy(1AS), asadmin-deploydir(1AS), asadmin-enable(1AS),
asadmin-disable(1AS), asadmin-list-components(1AS)

Sun ONE Application Server 7 man pages • Last Revised 9 Sep 2002
asadmin-unset(1AS)

NAME  
asadmin-unset, unset – removes one or more variables from the environment

SYNOPSIS  
unset env_var [env_var] *

DESCRIPTION  
Use the unset command to remove one or more variables you set for the environment. The variables and their associated values will no longer exist.

This command can be run remotely only.

OPERANDS  
env_var  
environment variable to be removed.

EXAMPLES  
EXAMPLE 1 Using unset to remove environment variables

asadmin> export AS_ADMIN_HOST=bluestar AS_ADMIN_PORT=8000
AS_ADMIN_USER=admin AS_ADMIN_PASSWORD=password
asadmin> export AS_ADMIN_PREFIX=server1.jms-service
asadmin> export
AS_ADMIN_HOST=bluestar
AS_ADMIN_PORT=8000
AS_ADMIN_USER=admin
AS_ADMIN_PASSWORD=********
AS_ADMIN_PREFIX=server1.jms-service
asadmin> unset AS_ADMIN_PREFIX
asadmin> export
AS_ADMIN_HOST=bluestar
AS_ADMIN_PORT=8000
AS_ADMIN_USER=admin
AS_ADMIN_PASSWORD=********

Using the export command without the argument lists the environment variables that are set. Notice the AS_ADMIN_PREFIX is not in the environment after running the unset command.

EXIT STATUS  
0  
command executed successfully

1  
error in executing the command

SEE ALSO  
asadmin-export(1AS), asadmin-multimode(1AS)
asadmin-update-file-user(1AS)

NAME
asadmin-update-file-user, update-file-user – updates a current file user as specified

SYNOPSIS
update-file-user --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename][--secure|-s][--instance instance_name] [--userpassword user_password][--groups user_groups:[user_groups]*] user_name

DESCRIPTION
Updates a current file user with the named items.

OPTIONS
--user
administrative user associated for the instance.

--password
administrative password corresponding to the administrative user.

--host
host name of the machine hosting the administrative instance.

--port
administrative port number associated with the administrative host.

--passwordfile
file containing passwords appropriate for the command (e.g., administrative instance).

--secure
if true, uses SSL/TLS to communicate with the administrative instance.

--instance
name of the instance.

--userpassword
password for the file user.

--groups
group where the file user belongs to.

OPERANDS
user_name
name of file user.

EXAMPLES
EXAMPLE 1 Using the update-file-user command to update a file user
asadmin> update-file-user --user admin --password admin_password
--host fuyako --port 7070 --instance server1 --userpassword sample_password
--groups staff:manager:engineer sample_user
Updated File user sample_user

Where: the sample_user is the file user updated with the updated user password and groups.

EXIT STATUS
0
command executed successfully

1
error in executing the command

SEE ALSO
asadmin-delete-file-user(1AS), asadmin-list-file-users(1AS),
asadmin-create-file-user(1AS), asadmin-list-file-groups(1AS)
NAME
asadmin-version, version – displays the version information for the Sun ONE Application Server

SYNOPSIS

DESCRIPTION
version displays the version information for the Sun ONE Application Server.

OPTIONS
--user administrative user associated for the instance.
--password administrative password corresponding to the administrative user.
--host host name of the machine hosting the administrative instance.
--port administrative port number associated with the administrative host.
--local determines if the command should delegate the request to the administrative instance or run locally.
--verbose displays version information in detail.
--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).
--secure if true, uses SSL/TLS to communicate with the administrative instance.

EXAMPLES
EXAMPLE 1 Using local mode to display version
asadmin> version
Sun ONE Application Server 7.0

EXAMPLE 2 Using local mode to display version in detail
asadmin> version --verbose --local
Sun ONE Application Server 7.0 (build A021930-126949)

EXAMPLE 3 Using remote mode to display version in detail
asadmin> version --user admin --password adminadmin --host bluestar port 4848 --verbose
Sun ONE Application Server 7.0 (build A021930-126949)

EXIT STATUS
0 command executed successfully
1 error in executing the command

SEE ALSO
asadmin-display-license(1AS)
### NAME
asant – launches the Jakarta Ant tool

### SYNOPSIS
`asant target_list`

### DESCRIPTION
Use the `asant` command to automate repetitive development and deployment tasks. `asant` is a shell script that invokes the underlying Ant infrastructure after initializing the environment to pickup the application server installed targets.

To use Ant as part of the Sun ONE Application Server, verify that your PATH includes the provided `asant` (Solaris) `ant.bat` (Windows) script.

The bundled sample applications use `asant` extensively; however, `asant` can be used in any development or operational environments.

The build targets are represented in the build.xml files that accompany the sample applications.

To use the Ant tool to compile and reassemble the sample applications, verify that the `$AS_INSTALL/bin` directory is on your environment’s path. On UNIX, add the `$AS_INSTALL/bin` directory to your PATH environment variable. On Windows, after installing the Sun ONE Application Server, set the system path by adding `$AS_INSTALL\bin` to the user PATH. You can access the PATH system variable from: Start menu, Settings, Control Panel, System, Advanced, Environment Variables, User Variables for Administrator, PATH.

The `target_list` is one or more space separated tasks as described below.

### TARGETS

- **compile** compiles all Java source code.
- **jar** assembles the EJB JAR module.
- **war** assembles the WAR file in `<sample_dir>/assemble/war`
- **ear** assembles the EAR file in `<sample_dir>/assemble/ear`
- **core** (default) compiles all sources, builds stubs and skeletons; and assembles EJB JAR, WAR and EAR files. This is the default target for all build.xml files shipped in the Sun ONE Application Server.
- **javadocs** creates Java docs in `<sample_dir>/javadocs`
- **all** builds core and javadocs, verifies and deploys the application, and adds the resources.
- **deploy** deploys the application and automatically expands the EJB JAR; does not install Javadocs.
- **undeploy** removes the deployed sample from the Sun ONE Application Server.
- **clean** removes `<appname>/build/` and `<appname>/assemble/` and `<appname>/javadocs` directories.
- **verify** verifies the deployment descriptors in the sample.
EXAMPLE 1 Compiling and Assembling a Sample Application

Using the simple stateless EJB sample as an example, execute several of the build targets as follows:

cd install_root/samples/ejb/stateless/simple/src

Execute the compile target to compile the Java sources as follows:

asant compile

Execute the war, ear, and ejbjar target to assemble the J2EE module files and the EAR file as follows by:

asant jar
asant war
asant ear

Alternatively, all the above tasks can be accomplished by:

asant core

Since the default build target is core you can execute asant without any arguments to rebuild the entire application.

EXAMPLE 2 Building Web-based Applications

You can build everything, including installing Javadocs, and deploying the application by:

asant all

Additionally, you can build everything, except the Javadocs, but deploy the application by:

asant core
or just,
asant then,
asant deploy

To rebuild the ear after you have modified the deployment descriptors without recompiling:

asant ear
asant deploy

SEE ALSO


SUNWWant documentation located in /usr/sfw/share/doc/ant
, asadmin(1AS)
**NAME**
capture-schema – stores the database metadata (schema) in a file for use in mapping and execution

**SYNOPSIS**
capture-schema -dburl url -username name - password password -driver a_jdbc_driver [-schemaname name] [-table tablename] * [-out filename]

**DESCRIPTION**
Use the capture-schema command to store the database metadata (schema) in a file. You can also use the Sun ONE Studio (formerly Forte for Java) IDE to capture the database schema.

**OPTIONS**
- **-dburl**
  JDBC URL expected by the driver for accessing a database.
- **-username**
  user name for authenticating access to a database.
- **-password**
  password for accessing the selected database.
- **-driver**
  JDBC driver classname in your CLASSPATH.
- **-schemaname**
  name of the user schema being captured. If not specified, the default will capture metadata for all tables from all the schemas accessible to this user. Specifying this parameter is highly recommended. If more than one schema is accessible to this user, more than one table with the same name may be captured which will cause problems.
- **-table**
  name of the table; multiple table names can be specified.
- **-out**
  output target; defaults to stdout. This parameter corresponds to the schema sub-element of the sun-cmp-mapping element in the sun-cmp-mapping_1_0.dtd file.

**EXAMPLES**
**EXAMPLE 1** Using capture-schema

**SEE ALSO**
asadmin(1AS)
NAME
flexanlg – analyzes access log files

SYNOPSIS
[-m metafile] * [-o filename] [-c options] [-t options] [-l options]

DESCRIPTION
Use the flexanlg command to generate statistics about your server, such as a
summary of activity, most commonly accessed URLs, times during the day when the
server is accessed most frequently, and so on.

These statistics are generated from the server’s access log which, by default, is named
access and is found in the logs directory of the server instance.

Before running the log analyzer, you should archive the server logs.

OPTIONS
Options marked with * can be repeated.

- i filename
  input log file(s)

- P
  proxy log format

- n servername
  the name of the server

- x
  output in HTML

- r
  resolve IP addresses to hostnames

- p [ c, t, l ]
  output order; default order is counts, time statistics, and lists

- m filename
  meta file(s)

- o filename
  output log file; default is stdout

- c [ h, n, r, f, e, u, o, k, c, z ]
  count these items; default is: h, n, r, e, u, o, k, c
  - h: total hits
  - n: 304 Not Modified status codes (use local copy)
  - r: 302 Found status codes (redirects)
  - f: 404 Not Found status codes (Document Not Found)
  - e: 500 Server Error status codes (Misconfiguration)
  - u: total unique URLs
  - o: total unique hosts
  - k: total kilobytes transferred
  - c: total kilobytes saved by caches
  - z: Do not count any items
flexanlg(1AS)

-t [ sx, mx, hx, xx, z ]
find general statistics; default is: s5m5h24x10
- s [number]: Find top (number) seconds of log
- m [number]: Find top (number) minutes of log
- h [number]: Find top (number) hours of log
- u [number]: Find top (number) users of log
- a [number]: Find top (number) user agents of log
- r [number]: Find top (number) referers of log
- x [number]: Find top (number) for miscellaneous keywords
- z: Do not find any general statistics

-l [ cx, hx ]
Make a list of the specified suboption; default is: c+3h5
- c [x, +x]: most commonly accessed URLs
  - x: only list x entries
  - +x: only list if accessed more than x times
- h [x, +x]: hosts or IP addresses most often accessing your server
  - x: only list x entries
  - +x: only list if accessed more than x times
- z: Do not make any lists

EXAMPLES
EXAMPLE 1 Using the flexanlg command
flexanlg -i /var/opt/SUNQappserver7/domains/domain1/server1/logs/access

SEE ALSO wscompile(1AS), wsdeploy(1AS)
<table>
<thead>
<tr>
<th>NAME</th>
<th>htpasswd – creates the user authentication files</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNOPSIS</td>
<td><strong>htpasswd</strong> [ -c ] passwdfile username</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Use the <strong>htpasswd</strong> utility to create the flat-files that store usernames and password for basic authentication of HTTP users. If <strong>htpasswd</strong> cannot access a file, such as not being able to write to the output file or not being able to read the file, it returns an error status and makes no changes.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td><code>-c</code> creates the <code>passwdfile</code>. If the <code>passwdfile</code> already exists, it is rewritten and truncated.</td>
</tr>
<tr>
<td></td>
<td><code>passwdfile</code> name of the file to contain the username and password.</td>
</tr>
<tr>
<td></td>
<td><code>username</code> the username to create in the <code>passwdfile</code>. If the username does not exist in this file, an entry is added. If it does exist, the password is changed.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td><strong>htpasswd</strong> -c myauthen scott</td>
</tr>
</tbody>
</table>
NAME  jspc – precompiles JSP source files into servlets

SYNOPSIS  jspc [options] jsp_files or jspc [options] -webapp dir

DESCRIPTION  Use the jspc command to compile your JSP 1.2 compliant source files into servlets. To allow the application server to pick up the precompiled JSPs from a JAR file, you must disable dynamic reloading of JSPs. To do this, set reload-interval property to -1 in the jsp-config element of the sun-web.xml file.

For more information about the sun-web.xml file, see the Sun ONE Application Server Developer’s Guide.

OPTIONS  

jsp_files  one or more JSP files to be compiled.

-webapp dir  a directory containing a web application. All JSPs in the directory and its subdirectories are compiled. You cannot specify a WAR, JAR, or ZIP file; you must first deploy it to an open directory structure using asadmin deploy.

-q  enables quiet mode (same as -v0). Only fatal error messages are displayed.

-d dir  the output directory for the compiled JSPs. Package directories are automatically generated based on the directories containing the uncompiled JSPs. The default top-level directory is the directory from which jspc is invoked.

-p name  the name of the target package for all specified JSPs, overriding the default package generation performed by the -d option.

-c name  the target class name of the first JSP compiled. Subsequent JSPs are unaffected.

-uriroot dir  the URI directory to which compilations are relative. Applies only to JSP files listed in the command, and not to JSP files specified with -webapp option. This is the location of each JSP file relative to the uriroot. If this cannot be determined, the default is /.  

-genclass  compiles the generated servlets into class files.
-v [level] enables verbose mode. The level is optional; the default is 2. Possible level values are:
  - 0 - fatal error messages only
  - 1 - error messages only
  - 2 - error and warning messages only
  - 3 - error, warning, and informational messages
  - 4 - error, warning, informational, and debugging messages

-mapped generates separate write calls for each HTML line and comments that describe the location of each line in the JSP file. By default, all adjacent write calls are combined and no location comments are generated.

-die [code] causes the JVM to exit and generates an error return code if a fatal error occurs. If the code is absent or unparsable it defaults to 1.

-webinc file creates partial servlet mappings for the -webapp option, which can be pasted into a web.xml file.

-webxml file creates an entire web.xml file for the -webapp option.

-ieplugin class_id specifies the Java plugin COM class ID for Internet Explorer. Used by the jsp:plugin tags.

EXAMPLES

EXAMPLE 1 Using jspc to compile the JSPs in a web application

The following command compiles a set of JSP files into Java files under Hellodir:

```
jspc -d Hellodir welcome.jsp shop.jsp checkout.jsp
```

The following command compiles all the JSP files in the specified webapp into class files under Hellodir:

```
jspc -d Hellodir -genclass -webapp /path_to_webapp_directory
```

To use these precompiled JSP in the web application, put the generated files under Hellodir into a JAR file, place the JAR file under WEB-INF/lib and set reload-interval property to -1 in the jsp-config element of the WEB-INF/sun-web.xml file.

SEE ALSO asadmin(1AS)
package-appclient – packs the application client container libraries and jar files

SYNOPSIS

package-appclient

DESCRIPTION

Use the package-appclient command to pack the application client container libraries and jar files into an appclient.jar file. The created file is located at appserver_install_dir/lib/appclient/appclient.jar. The appclient.jar file provides an application client container package targeted at remote hosts that do not contain a server installation.

The appclient.jar file contains native code and can be used on a target machine that is of similar architecture as the machine where it was produced. So, for example, an appclient.jar produced on a Solaris SPARC platform cannot be used on a Windows client machine.

After copying the appclient.jar file to a remote location, unjar it to get a set of libraries and jar files in the appclient directory.

After unjarring on the client machine, modify appclient_install_dir/config/asenv.conf (asenv.bat for Windows) as follows:

- set AS_WEBSERVICES_LIB to appclient_install_dir/lib
- set AS_NSS to appclient_install_dir/lib (appclient_install_dir/bin for Windows)
- set AS_IMQ_LIB to appclient_install_dir/imq/lib
- set AS_INSTALL to appclient_install_dir
- set AS_JAVA to your JDK 1.4 home directory
- set AS_ACC_CONFIG to appclient_install_dir/config/sun-acc.xml

Modify appclient_install_dir/config/sun-acc.xml as follows:

- Ensure the DOCTYPE file references appclient_install_dir/lib/dtds
- Ensure that target-server address attribute references the server machine.
- Ensure that target-server port attribute references the ORB port on the remote machine.
- Ensure that log-service references a log file; if the user wants to put log messages to a log file.

Modify appclient_install_dir/bin/appclient (appclient.bat for Windows) as follows:

- change token %CONFIG_HOME% to appclient_install_dir/config

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Unstable</td>
</tr>
</tbody>
</table>
SEE ALSO | appclient (IAS)
Use the verifier utility to validate the J2EE deployment descriptors and the Sun ONE Application Server specific deployment descriptors. If the application is not J2EE compliant, an error message is printed.

When you run the verifier utility, two results files are created in XML and TXT format. The location where the files are created can be configured using the -d option. The directory specified as the destination directory for result files should exist. If no directory is specified, the result files are created in the current directory. Result files are named as `jar_filename_verified.xml` and `jar_filename_verified.txt`.

The XML file has various sections that are dynamically generated depending on what kind of application or module is being verified. The root tag is `static-verification` which may contain the tags `application`, `ejb`, `web`, `appclient`, `connector`, `other`, `error` and `failure-count`. The tags are self explanatory and are present depending on the type of module being verified. For example, an EAR file containing a web and EJB module will contain the tags `application`, `ejb`, `web`, `other`, and `failure-count`.

If the verifier ran successfully, a result code of 0 is returned. A non-zero error code is returned if the verifier failed to run.

**OPTIONS**

- `-v`   verbose debugging is turned on.
- `-d`   identifies where the result files get placed.
- `-r`   identifies the reporting level defined as one of the following:
  - `a` sets output reporting level to display all results (default)
  - `w` sets output reporting level to display warning and failure results
  - `f` sets output reporting level to display only failure results

`jar_filename`   name of the ear/war/jar file to perform static verification on. The results of verification are placed in two files `jar_filename_verified.xml` and `jar_filename_verified.txt` in the destination directory.

**EXAMPLES**

**EXAMPLE 1** Using verifier in the Verbose Mode

```
example% verifier -v -d /verifier-results -rf sample.ear
```

Where `-v` runs the verifier in verbose mode, `-d` specifies the destination directory, and `-rf` displays only the failures. The results are stored in `/verifier-results/sample.ear_verified.xml` and `/verifier-results/sample.ear_verified.txt`.

**SEE ALSO**

asadmin(1AS)
NAME
wscompile – generates stubs, ties, serializers, and WSDL files used in JAX-RPC clients and services

SYNOPSIS
wscompile [options]configuration_file

DESCRIPTION
Use the wscompile command to generate the client stubs and server-side ties for the service definition interface that represents the web service interface. Additionally, use the wscompile command to generate the WSDL description of the web service interface which is then used to generate the implementation artifacts.

In addition to supporting the generation of stubs, ties, server configuration, and WSDL documents from a set of RMI interfaces, wscompile also supports generating stubs, ties and remote interfaces from a WSDL document.

You must specify one of the -gen options in order to use wscompile as a stand alone generator. You must use either -import (for WSDL) or -define (for an RMI interface) along with the -model option in order to use wscompile in conjunction with wsdeploy.

Invoking the wscompile command without specifying any arguments outputs the usage information.

OPTIONS
- cp path location of the input class files.
- classpath path same as -cp path option.
- d directory where to place the generated output files.
- define read the service’s RMI interface, define a service. Use this option with the -model option in order to create a model file for use with the wsdeploy command.
- f:features enables the given features. Features are specified as a comma separated list of features. See the list of supported features below.
- features:features same as -f:features option.
- g generates the debugging information.
- gen generates the client-side artifacts.
- gen:client same as -gen option.
- gen:server generates the server-side artifacts.
- gen:both generates client and server artifacts.
- httpproxy:host:port specifies a HTTP proxy server; defaults to port 8080.
- import reads a WSDL file, generates the service’s RMI interface and a template of the class that implements the interface. Use this option with the -model option in order to create a model file for use with the wsdeploy command.
wscompile(1AS)

-`model` write the internal model for the given file name. Use this option with the -`import` option in order to create a model file for use with the `wsdeploy` command.

-`-keep` keeps the generated files.

-`-nd directory` directory for the non-class generated files.

-`-O` optimizes the generated code.

-`-s directory` directory for the generated source files.

-`-verbose` outputs messages about what the compiler is doing.

-`-version` prints version information.

Exactly one of the -`input`, -`define`, -`gen` options must be specified.

-`datahandleronly` always map attachments to data handler type

-`explicitcontext` turn on explicit service context mapping.

-`infix=name` specify an infix to use for generated serializers.

-`nodatabinding` turn off data binding for literal encoding.

-`noencodedtypes` turn off encoding type information.

-`nomultirefs` turn off support for multiple references.

-`novalidation` turn off validation for the imported WSDL file.

-`searchschema` search schema aggressively for subtypes.

-`serializeinterfaces` turn on direct serialization of interface types.

Note: the -`gen` options are not compatible with `wsdeploy`.

**SUPPORTED FEATURES**

**CONFIGURATION FILE FORMAT**

The `wscompile` commands reads the configuration file `config.xml` which contains information that describes the web service. The structure of the file is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<configuration
xmlns="http://java.sun.com/xml/ns/jax-rpc/ri/config">
<service> or <wsdl> or <modelfile>
</configuration>
```

The configuration element may contain exactly one `<service>`, `<wsdl>` or `<modelfile>`. 
If the `<service>` element is specified, `wscompile` reads the RMI interface that describes the service and generates a WSDL file. In the `<interface>` subelement, the `name` attribute specifies the service’s RMI interface, and the `servantName` attribute specifies the class that implements the interface. For example:

```xml
<service name="CollectionIF_Service"
    targetNamespace="http://echoservice.org/wsdl"
    typeNamespace="http://echoservice.org/types"
    packageName="stub_tie_generator_test">
    <interface name="stub_tie_generator_test.CollectionIF"
        servantName="stub_tie_generator_test.CollectionImpl"/>
</service>
```

If the `<wsdl>` element is specified, `wscompile` reads the WSDL file and generates the service’s RMI interface. The `location` attribute specifies the URL of the WSDL file, and the `packageName` attribute specifies the package of the classes to be generated. For example:

```xml
<wsdl
    location="http://tempuri.org/sample.wsdl"
    packageName="org.tempuri.sample"/>
```

If `config.xml` contains a `<service>` or `<wsdl>` element, `wscompile` can generate a model file that contains the internal data structures that describe the service. If a model file is already generated, it can be reused next time while using `wscompile`. For example:

```xml
<modelfile location="mymodel.Z"/>
```

**EXAMPLE 1** Using `wscompile` to generate client-side artifacts

```
wscompile -gen:client -d outputdir -classpath classpathdir config.xml
```

Where a client side artifact is generated in the `outputdir` for running the service as defined in the `config.xml` file.

**EXAMPLE 2** Using `wscompile` to generate server-side artifacts

```
wscompile -gen:server -d outputdir -classpath classpathdir -model modelfile.Z config.xml
```

Where a server side artifact is generated in the `outputdir` and the `modelfile` in `modelfile.Z` for services defined in the `config.xml` file.
wscompile(1AS)

SEE ALSO  
wsdeploy(1AS)
wsdeploy reads a WAR file and the jaxrpc-ri.xml file and generates another WAR file that is ready for deployment.

Use the `wsdeploy` command to take a WAR file which does not have implementation specific server side tie classes to generate a deployable WAR file that can be deployed on the application server. `wsdeploy` internally runs `wscompile` with the `-gen:server` option. The `wscompile` command generates classes and a WSDL file which `wsdeploy` includes in the generated WAR file.

**OPTIONS**

- **-classpath path** location of the input class files.
- **-keep** keep temporary files.
- **-tmpdir** temporary directory to use.
- **-o output WAR file** required; location of the generated WAR file.
- **-verbose** outputs messages about what the compiler is doing.
- **-version** prints version information.

**EXAMPLES**

The input WAR file for `wsdeploy` will typically have the following structure:

```
META-INF/
MANIFEST.MF
WEB-INF/
    web.xml (normal deployment descriptor)
    jaxrpc-ri.xml
    <modelfile>.Z (optional model file generated using `wscompile` with `-model` option)
classes/
    Your application
```

Running `wsdeploy` on the above WAR file results in the following actions:

```
web.xml
    Renames to web-before.xml
    Adds elements: listener, servlet, and servlet-mapping
jaxrpc-ri.xml
    renames to jaxrpc-before.xml
    writes jaxrpc-runtime.xml
    A list of endpoints containing: name, interface, implementation, tie, model, WSDL, service, port, and URL pattern attributes
Generates classes and WSDL using `wscompile -gen:server`
Packages the output.war file
```

**EXAMPLE 1** Creating the output WAR file

```
wsdeploy -o Hello.war Input.war
```

Where the deployable WAR file `Hello.war` is generated.
EXAMPLE 1 Creating the output WAR file  (Continued)

The contents of the Input.war for a simple Hello web service would be:

META-INF/MANIFEST.MF
WEB-INF/classes/hello/HelloIF.class
WEB-INF/classes/hello/HelloImpl.class
WEB-INF/jaxrpc-ri.xml
WEB-INF/web.xml

Where:

HelloIF is the service’s RMI interface
HelloImpl is the class that implements the interface
web.xml file is the deployment descriptor of a web component

The contents of the jaxrpc-ri.xml file would be:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<webServices
xmlns="http://java.sun.com/xml/ns/jax-rpc/ri/dd"
version="1.0"
targetNamespaceBase="http://com.test/wsdl"
typeNamespaceBase="http://com.test/types"
urlPatternBase="/ws">
<endpoint
name="MyHello"
displayName="HelloWorld Service"
description="A simple web service"
interface="hello.HelloIF"
implementation="hello.HelloImpl"/>
<endpointMapping
wsdeploy(1AS)
EXAMPLE 1 Creating the output WAR file (Continued)

    <endpointName="MyHello"
    urlPattern="/hello"/>
</webServices>

SEE ALSO wscompile(1AS)
Index

a
analyzes access log files — flexanlg, 175

r
reads a WAR file and the jaxrpc-ri.xml file
and generates another WAR file that is ready
for deployment — wsdeploy, 187

A
adds a lifecycle module for the named instance
— asadmin-create-lifecycle-module, 71
adds a lifecycle module for the named instance
— create-lifecycle-module, 71
adds a new access control list file for the named
instance — asadmin-create-acl, 40
adds a new access control list file for the named
instance — create-acl, 40
adds a new HTTP listener socket —
asadmin-create-http-listener, 49
adds a new HTTP listener socket —
create-http-listener, 49
adds the IIOP listener for the named instance
— asadmin-create-iop-listener, 53
adds the IIOP listener for the named instance
— create-iop-listener, 53
adds the MIME type for the named instance
— asadmin-create-mime, 73
adds the MIME type for the named instance
— create-mime, 73
adds the named destination —
asadmin-create-jmsdest, 64
adds the named destination —
create-jmsdest, 64
adds the named virtual server —
asadmin-create-virtual-server, 80
adds the named virtual server —
create-virtual-server, 80
adds the new authorized database for the
named instance — asadmin-create-
authdb, 41
adds the new authorized database for the
named instance — create-authdb, 41
adds the new authorized realm for the named
instance — asadmin-create-auth-realm, 43
adds the new authorized realm for the named
instance — create-auth-realm, 43
allows you to execute multiple commands
while returning environment settings and
remaining in the asadmin utility —
asadmin-multimode, 149
allows you to execute multiple commands
while returning environment settings and
remaining in the asadmin utility —
multimode, 149
appclient — launches the Application Client
Container and invokes the client application
packaged in the application JAR file, 16
applies the changes you have made for a server
instance — asadmin-reconfig, 150
applies the changes you have made for a server
instance — reconfig, 150
asadmin — utility for performing administrative tasks for the Sun ONE Application Server, 18
asadmin-create-acl — adds a new access control list file for the named instance, 40
asadmin-create-auth-realm — adds the new authorized realm for the named instance, 43
asadmin-create-authdb — adds the new authorized database for the named instance, 41
asadmin-create-domain — creates a domain with the given name, 46
asadmin-create-file-user — creates a new file user, 47
asadmin-create-http-listener — adds a new HTTP listener socket, 49
asadmin-create-http-qos — creates a new quality of service parameter for the named instance, 51
asadmin-create-iiop-listener — adds the IIOP listener for the named instance, 53
asadmin-create-instance — creates an application server instance with the specified instance name, 55
asadmin-create-javamail-resource — registers the Javamail resource to the named instance, 57
asadmin-create-jdbc-connection-pool — registers the JDBC connection pool to the named instance, 59
asadmin-create-jdbc-resource — registers the JDBC resource to the named instance, 62
asadmin-create-jms-resource — registers the JMS resource to the named instance, 65
asadmin-create-jmsdest — adds the named destination, 64
asadmin-create-jndi-resource — registers the JNDI resource to the named instance, 67
asadmin-create-jvm-options — creates the JVM options from the Java configuration or profiler elements, 69
asadmin-create-lifecycle-module — adds a lifecycle module for the named instance, 71
asadmin-create-mime — adds the MIME type for the named instance, 73
asadmin-create-persistence-resource — registers the persistence resource to the named instance, 74
asadmin-create-profiler — creates the profiler element, 76
asadmin-create-ssl — creates the SSL element in the HTTP listener or IIOP listener, 78
asadmin-create-virtual-server — adds the named virtual server, 80
asadmin-delete-acl — removes the access control list file for the named instance, 82
asadmin-delete-auth-realm — removes the named authorized realm, 84
asadmin-delete-authdb — removes the authorized database for the named instance, 83
asadmin-delete-custom-resource — removes the custom resource from the named instance, 85
asadmin-delete-domain — deletes the given domain, 86
asadmin-delete-file-user — removes the named file user, 87
asadmin-delete-http-listener — removes the HTTP listener for the named instance, 88
asadmin-delete-http-qos — removes the quality of service parameter for the named instance, 89
asadmin-delete-iiop-listener — removes the IIOP listener for the named instance, 90
asadmin-delete-instance — deletes the instance that is not running, 91
asadmin-delete-javamail-resource — removes the Javamail resource from the named instance, 93
asadmin-delete-jdbc-connection-pool — removes the JDBC connection pool from the named instance, 94
asadmin-delete-jdbc-resource — removes the JDBC resource from the named instance, 95
asadmin-delete-jms-resource — removes the JMS resource from the named instance, 97
asadmin-delete-jmsdest — destroys the named destination, 96
asadmin-delete-jndi-resource — removes the JNDI resource from the named instance, 98
asadmin-delete-jvm-options — deletes the JVM options from the Java configuration or profiler elements, 99
asadmin-delete-lifecycle-module — removes the lifecycle module for the named instance, 101
asadmin-delete-mime — removes the MIME type for the named instance, 102
asadmin-delete-persistence-resource — removes the persistence resource from the named instance, 103
asadmin-delete-profiler — deletes the profiler element, 104
asadmin-delete-virtual-server — deletes the virtual server with the named virtual server ID, 106
asadmin-deploy — deploys the specified component, 107
asadmin-deploydir — deploys the J2EE component that is in the directory located on the server machine, 109
asadmin-disable — stops the specified component, 111
asadmin-display-license — displays the license information, 112
asadmin-enable — runs the specified component, 113
asadmin-export — marks a variable name for automatic export to the environment of subsequent commands in multimode, 114
asadmin-get — gets the values of the monitorable or configurable attributes, 115
asadmin-help — displays a list of all the commands available in the Command-line interface, 117
asadmin-install-license — installs the license file, 120
asadmin-jms-ping — checks to see if the JMS provider is up and running, 121
asadmin-list — lists the configurable elements, 122
asadmin-list-acls — gets the access control lists for the named instance, 124
asadmin-list-auth-realms — lists the authorized realms associated with the named instance, 126
asadmin-list-authdbs — gets the authorized database for the named instance, 125
asadmin-list-components — lists deployed J2EE components, 127
asadmin-list-custom-resources — gets all the custom resources from the named instance, 129
asadmin-list-domains — lists all the domains, 130
asadmin-list-file-groups — lists the file groups for the named instance, 131
asadmin-list-http-listeners — gets the HTTP listeners for the named instance, 133
asadmin-list-iop-listeners — gets the IIOp listeners for the named instance, 134
asadmin-list-instances — lists all the instances in the server, 135
asadmin-list-javamail-resources — gets all the Javamail resources from the named instance, 137
asadmin-list-jms-resources — gets all the JMS resources from the named instance, 141
asadmin-list-jmsdest — gets all the named destinations, 140
asadmin-list-jndi-resources — gets all the JNDI resources from the named instance, 142
asadmin-list-mimes — gets the MIME types for the named instance, 144
asadmin-list-persistence-resources — gets all the persistence resources from the named instance, 145
asadmin-list-sub-components — lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application, 147
asadmin-list-virtual-servers — gets the virtual servers in the named instance, 148
asadmin-multimode — allows you to execute multiple commands while returning environment settings and remaining in the asadmin utility, 149
asadmin-reconfig — applies the changes you have made for a server instance, 150
asadmin-restart-instance — restarts the specified server instance and all the services associated with it, 152
asadmin-set — sets the values of attributes, 154
asadmin-show-component-status — displays the status of the deployed component, 155
asadmin-show-instance-status — displays the status of the server instance specified, 157
asadmin-shutdown — brings down the administration server and associated instances, 158
asadmin-start-appserv — starts the local Administration server and all the instances associated with it, 159
asadmin-start-instance — starts a server instance and all the services associated with it, 161
asadmin-stop-appserv — stops the local administration server and all the instances associated with it, 163
asadmin-stop-domain — stops the given domain, 164
asadmin-stop-instance — stops the specified server instance and all the services associated with it, 166
asadmin-undeploy — removes the component from the named instance, 168
asadmin-unset — removes one or more variables from the environment, 169
asadmin-update-file-user — updates a current file user as specified, 170
asadmin-version — displays the version information for the Sun ONE Application Server, 171
asant — launches the Jakarta Ant tool, 172

B
brings down the administration server and associated instances — asadmin-shutdown, 158
brings down the administration server and associated instances — shutdown, 158

C
capture-schema — stores the database metadata (schema) in a file for use in mapping and execution, 174
checks to see if the JMS provider is up and running — asadmin-jms-ping, 121
checks to see if the JMS provider is up and running — jms-ping, 121
create-acl — adds a new access control list file for the named instance, 40
create-auth-realm — adds the new authorized realm for the named instance, 43
create-authdb — adds the new authorized database for the named instance, 41
create-domain — creates a domain with the given name, 46
create-file-user — creates a new file user, 47
create-http-listener — adds a new HTTP listener socket, 49
create-http-qos — creates a new quality of service parameter for the named instance, 51
create-iiop-listener — adds the IIOP listener for the named instance, 53
create-instance — creates an application server instance with the specified instance name, 55
create-javamail-resource — registers the Javamail resource to the named instance, 57
create-jdbc-connection-pool — registers the JDBC connection pool to the named instance, 59
create-jdbc-resource — registers the JDBC resource to the named instance, 62
create-jms-resource — registers the JMS resource to the named instance, 65
create-jmsdest — adds the named destination, 64
create-jndi-resource — registers the JNDI resource to the named instance, 67
create-jvm-options — creates the JVM options from the Java configuration or profiler elements, 69
create-lifecycle-module — adds a lifecycle module for the named instance, 71
create-mime — adds the MIME type for the named instance, 73
create-persistence-resource — registers the persistence resource to the named instance, 74
create-profiler — creates the profiler element, 76
create-ssl — creates the SSL element in the HTTP listener or IIOP listener, 78
create-virtual-server — adds the named virtual server, 80
creates a domain with the given name — asadmin-create-domain, 46
creates a domain with the given name — create-domain, 46
creates a new file user — asadmin-create-file-user, 47
creates a new file user — create-file-user, 47
creates a new quality of service parameter for the named instance — asadmin-create-http-qos, 51
creates a new quality of service parameter for the named instance — create-http-qos, 51
creates an application server instance with the specified instance name — asadmin-create-instance, 55
creates an application server instance with the specified instance name — create-instance, 55
creates the JVM options from the Java configuration or profile elements — asadmin-create-jvm-options, 69
creates the JVM options from the Java configuration or profile elements — create-jvm-options, 69
creates the profiler element — asadmin-create-profiler, 76
creates the profiler element — create-profiler, 76
creates the SSL element in the HTTP listener or IIOP listener — asadmin-create-ssl, 78
creates the SSL element in the HTTP listener or IIOP listener — create-ssl, 78
creates the user authentication files — htpasswd, 177

D
delete-acl — removes the access control list file for the named instance, 82
delete-auth-realm — removes the named authorized realm, 84
delete-authdb — removes the authorized database for the named instance, 83
delete-custom-resource — removes the custom resource from the named instance, 85
delete-domain — deletes the given domain, 86
delete-file-user — removes the named file user, 87
delete-http-listener — removes the HTTP listener for the named instance, 88
delete-http-qos — removes the quality of service parameter for the named instance, 89
delete-iiopt-listener — removes the IIOP listener for the named instance, 90
delete-instance — deletes the instance that is not running, 91
delete-javamail-resource — removes the Javamail resource from the named instance, 93
delete-jdbc-connection-pool — removes the JDBC connection pool from the named instance, 94
delete-jdbc-resource — removes the JDBC resource from the named instance, 95
delete-jms-resource — removes the JMS resource from the named instance, 97
delete-jmsdestination — destroys the named destination, 96
delete-jndi-resource — removes the JNDI resource from the named instance, 98
delete-jvm-options — deletes the JVM options from the Java configuration or profile elements, 99
delete-lifecycle-module — removes the lifecycle module for the named instance, 101
delete-mime — removes the MIME type for the named instance, 102
delete-persistence-resource — removes the persistence resource from the named instance, 103
delete-profiler — deletes the profiler element, 104
delete-virtual-server — deletes the virtual server with the named virtual server ID, 106
deletes the given domain — asadmin-delete-domain, 86
deletes the given domain — delete-domain, 86
deletes the instance that is not running — asadmin-delete-instance, 91
deletes the instance that is not running — delete-instance, 91
deletes the JVM options from the Java configuration or profile elements — asadmin-delete-jvm-options, 99
deletes the JVM options from the Java configuration or profile elements — delete-jvm-options, 99
deletes the profiler element —
   asadmin-delete-profiler, 104
deletes the profiler element —
   delete-profiler, 104
deletes the virtual server with the named virtual server ID — asadmin-delete-virtual-server, 106
deletes the virtual server with the named virtual server ID — delete-virtual-server, 106
deploy — deploys the specified component, 107
deploydir — deploys the J2EE component that is in the directory located on the server machine, 109
deploys the J2EE component that is in the directory located on the server machine —
   asadmin-deploydir, 109
deploys the J2EE component that is in the directory located on the server machine —
   deploydir, 109
deploys the specified component —
   asadmin-deploy, 107
deploys the specified component —
   deploy, 107
destroys the named destination —
   asadmin-delete-jmsdest, 96
destroys the named destination —
   delete-jmsdest, 96
disable — stops the specified component, 111
display-license — displays the license information, 112
displays a list of all the commands available in the Command-line interface —
   asadmin-help, 117
displays a list of all the commands available in the Command-line interface — help, 117
displays the license information —
   asadmin-display-license, 112
displays the license information —
   display-license, 112
displays the status of the server instance specified — asadmin-show-instance-status, 157
displays the status of the server instance specified — show-instance-status, 157
displays the version information for the Sun ONE Application Server —
   asadmin-version, 171
displays the version information for the Sun ONE Application Server — version, 171
displays the status of the deployed component —
   asadmin-show-component-status, 155
displays the status of the deployed component —
   show-component-status, 155

E
enable — runs the specified component, 113
export — marks a variable name for automatic export to the environment of subsequent commands in multimode, 114

F
flexanlg — analyzes access log files, 175

G
get — gets the values of the monitorable or configurable attributes, 115
gets all the custom resources from the named instance — asadmin-list-custom-resources, 129
gets all the custom resources from the named instance — list-custom-resources, 129
gets all the Javamail resources from the named instance — asadmin-list-javamail-resources, 137
gets all the Javamail resources from the named instance — list-javamail-resources, 137
gets all the JMS resources from the named instance — asadmin-list-jms-resources, 141
gets all the JMS resources from the named instance — list-jms-resources, 141
gets all the JNDI resources from the named instance — asadmin-list-jndi-resources, 142
gets all the JNDI resources from the named instance — list-jndi-resources, 142
gets all the named destinations —
   asadmin-list-jmsdest, 140
gets all the named destinations —
list-jmsdest, 140
gets all the persistence resources from the
named instance — asadmin-list-persistence-
resources, 145
gets all the persistence resources from the
named instance — list-persistence-
resources, 145
gets the access control lists for the named
instance — asadmin-list-acls, 124
gets the access control lists for the named
instance — list-acls, 124
gets the authorized database for the named
instance — asadmin-list-authdbs, 125
gets the authorized database for the named
instance — list-authdbs, 125
gets the HTTP listeners for the named instance
— asadmin-list-http-listeners, 133
gets the HTTP listeners for the named instance
— list-http-listeners, 133
gets the IIOP listeners for the named instance —
asadmin-list-iiop-listeners, 134
gets the IIOP listeners for the named instance —
list-iiop-listeners, 134
gets the MIME types for the named instance —
asadmin-list-mimes, 144
gets the MIME types for the named instance —
list-mimes, 144
gets the values of the monitorable or
configurable attributes — get, 115
gets the values of the monitorable or
configurable attributes. — asadmin-get, 115
gets the virtual servers in the named instance —
asadmin-list-virtual-servers, 148
gets the virtual servers in the named instance —
list-virtual-servers, 148

H help — displays a list of all the commands
available in the Command-line interface, 117
htpasswd — creates the user authentication
files, 177

I install-license — installs the license file, 120
installs the license file — asadmin-install-
license, 120
installs the license file — install-license, 120

J jms-ping — checks to see if the JMS provider is
up and running, 121
jspc — precompiles JSP source files into
Servlets, 178

L launches the Application Client Container and
invokes the client application packaged in
the application JAR file — appclient, 16
launches the Jakarta Ant tool — asant, 172
list — lists the configurable elements, 122
list-acls — gets the access control lists for the
named instance, 124
list-auth-realms — lists the authorized realms
associated with the named instance, 126
list-authdbs — gets the authorized database for
the named instance, 125
list-components — lists deployed J2EE
components, 127
list-custom-resources — gets all the custom
resources from the named instance, 129
list-domains — lists all the domains, 130
list-file-groups — lists the file groups for the
named instance, 131
list-http-listeners — gets the HTTP listeners for
the named instance, 133
list-iiop-listeners — gets the IIOP listeners for
the named instance, 134
list-instances — lists all the instances in the
server, 135
list-javamail-resources — gets all the Javamail
resources from the named instance, 137
list-jms-resources — gets all the JMS resources
from the named instance, 141
list-jmsdest — gets all the named
destinations, 140
list-jndi-resources — gets all the JNDI resources from the named instance, 142
list-mimes — gets the MIME types for the named instance, 144
list-persistence-resources — gets all the persistence resources from the named instance, 145
list-sub-components — lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application, 147
list-virtual-servers — gets the virtual servers in the named instance, 148
lists all the domains — asadmin-list-domains, 130
lists all the domains — list-domains, 130
lists all the instances in the server — asadmin-list-instances, 135
lists all the instances in the server — list-instances, 135
lists deployed J2EE components — asadmin-list-components, 127
lists deployed J2EE components — list-components, 127
lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application — asadmin-list-sub-components, 147
lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application — list-sub-components, 147
lists the authorized realms associated with the named instance — asadmin-list-auth-realms, 126
lists the authorized realms associated with the named instance — list-auth-realms, 126
lists the configurable elements — asadmin-list, 122
lists the configurable elements — list, 122
lists the file groups for the named instance — asadmin-list-file-groups, 131
lists the file groups for the named instance — list-file-groups, 131

M
marks a variable name for automatic export to the environment of subsequent commands in multimode — asadmin-export, 114
marks a variable name for automatic export to the environment of subsequent commands in multimode — export, 114
multimode — allows you to execute multiple commands while returning environment settings and remaining in the asadmin utility, 149

P
package-appclient — packs the application client container libraries and jar files, 180
packages the application client container libraries and jar files — package-appclient, 180
precompiles JSP source files into servlets — jspc, 178

R
reconfig — applies the changes you have made for a server instance, 150
registers the JavaMail resource to the named instance — asadmin-create-javamail-resource, 57
registers the JavaMail resource to the named instance — create-javamail-resource, 57
registers the JDBC connection pool to the named instance — asadmin-create-jdbc-connection-pool, 59
registers the JDBC connection pool to the named instance — create-jdbc-connection-pool, 59
registers the JDBC resource to the named instance — asadmin-create-jdbc-resource, 62
registers the JDBC resource to the named instance — create-jdbc-resource, 62
registers the JMS resource to the named instance — asadmin-create-jms-resource, 65
registers the JMS resource to the named instance — create-jms-resource, 65
registers the JNDI resource to the named instance — asadmin-create-jndi-resource, 67
registers the JNDI resource to the named instance — create-jndi-resource, 67
registers the persistence resource to the named instance — asadmin-create-persistence-resource, 74
registers the persistence resource to the named instance — create-persistence-resource, 74
removes one or more variables from the environment — asadmin-unset, 169
removes one or more variables from the environment — unset, 169
removes the access control list file for the named instance — asadmin-delete-acl, 82
removes the access control list file for the named instance — delete-acl, 82
removes the authorized database for the named instance — asadmin-delete-authdb, 83
removes the authorized database for the named instance — delete-authdb, 83
removes the component from the named instance — asadmin-undeploy, 168
removes the component from the named instance — undeploy, 168
removes the custom resource from the named instance — asadmin-delete-custom-resource, 85
removes the custom resource from the named instance — delete-custom-resource, 85
removes the HTTP listener for the named instance — asadmin-delete-http-listener, 88
removes the HTTP listener for the named instance — delete-http-listener, 88
removes the IIOP listener for the named instance — asadmin-delete-iiop-listener, 90
removes the IIOP listener for the named instance — delete-iiop-listener, 90
removes the Javamail resource from the named instance — asadmin-delete-javamail-resource, 93
removes the Javamail resource from the named instance — delete-javamail-resource, 93
removes the JDBC connection pool from the named instance — asadmin-delete-jdbc-connection-pool, 94
removes the JDBC connection pool from the named instance — delete-jdbc-connection-pool, 94
removes the JDBC resource from the named instance — asadmin-delete-jdbc-resource, 95
removes the JDBC resource from the named instance — delete-jdbc-resource, 95
removes the JMS resource from the named instance — asadmin-delete-jms-resource, 97
removes the JMS resource from the named instance — delete-jms-resource, 97
removes the JNDI resource from the named instance — asadmin-delete-jndi-resource, 98
removes the JNDI resource from the named instance — delete-jndi-resource, 98
removes the lifecycle module for the named instance — asadmin-delete-lifecycle-module, 101
removes the lifecycle module for the named instance — delete-lifecycle-module, 101
removes the MIME type for the named instance — asadmin-delete-mime, 102
removes the MIME type for the named instance — delete-mime, 102
removes the named authorized realm — asadmin-delete-auth-realm, 84
removes the named authorized realm — delete-auth-realm, 84
removes the named file user — asadmin-delete-file-user, 87
removes the named file user — delete-file-user, 87
removes the persistence resource from the named instance — asadmin-delete-persistence-resource, 103
removes the persistence resource from the named instance — delete-persistence-resource, 103
removes the quality of service parameter for the named instance — asadmin-delete-http-qos, 89
removes the quality of service parameter for the named instance — delete-http-qos, 89
restart-instance — restarts the specified server instance and all the services associated with it, 152
restarts the specified server instance and all the services associated with it — asadmin-restart-instance, 152
restarts the specified server instance and all the
services associated with it —
restart-instance, 152
runs the specified component —
asadmin-enable, 113
runs the specified component — enable, 113

S
set — sets the values of attributes, 154
sets the values of attributes — asadmin-set, 154
sets the values of attributes — set, 154
show-component-status — displays the status
of the deployed component, 155
show-instance-status — displays the status of
the server instance specified, 157
shutdown — brings down the administration
server and associated instances, 158
start-appserv — starts the local Administration
server and all the instances associated with
it, 159
start-instance — starts a server instance and all
the services associated with it, 161
starts a server instance and all the services
associated with it — asadmin-start-
instance, 161
starts a server instance and all the services
associated with it — start-instance, 161
starts the local Administration server and all the
instances associated with it —
asadmin-start-appserv, 159
starts the local Administration server and all the
instances associated with it —
start-appserv, 159
stop-appserv — stops the local administration
server and all the instances associated with
it, 163
stop-domain — stops the given domain, 164
stop-instance — stops the specified server
instance and all the services associated with
it, 166
stops the given domain —
asadmin-stop-domain, 164
stops the given domain — stop-domain, 164
stops the specified component —
asadmin-disable, 111
stops the specified component — disable, 111

V
validates the J2EE Deployment Descriptors
against application server DTDs —
verifier, 182
verifier — validates the J2EE Deployment
Descriptors against application server
DTDs, 182
version — displays the version information for
the Sun ONE Application Server, 171

 Sun customer support, 12

Sun ONE Application Server 7 man pages
wsdeploy — reads a WAR file and the
jaxrpc-ri.xml file and generates another
WAR file that is ready for deployment, 187