



Sun Java System Application Server Man Pages section 1M: Utility Commands

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

Part No: 817-5053
April 2004

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

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

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

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

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

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions.

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

Copyright 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

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

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

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

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

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



040407 @ 8606



Contents

Preface 7

User Commands 11

add-iiop-cluster-endpoint(1)	12
add-resources(1)	14
appclient(1M)	15
asadmin(1M)	17
asant(1M)	38
capture-schema(1M)	40
cladmin(1M)	41
clear-session-store(1)	45
clsetup(1M)	46
configure-session-persistence(1)	52
create-acl(1)	55
create-authdb(1)	56
create-auth-realm(1)	58
create-custom-resource(1)	59
create-domain(1)	61
create-file-user(1)	62
create-http-listener(1)	64
create-http-qos(1)	66
create-iiop-listener(1)	68
create-instance(1)	70
create-javamail-resource(1)	72
create-jdbc-connection-pool(1)	74
create-jdbc-resource(1)	77

create-jmsdest(1) 78
 create-jms-resource(1) 79
 create-jndi-resource(1) 81
 create-jvm-options(1) 83
 create-lifecycle-module(1) 85
 create-mime(1) 87
 create-persistence-resource(1) 88
 create-profiler(1) 90
 create-session-store(1) 92
 create-ssl(1) 93
 create-virtual-server(1) 95
 delete-acl(1) 97
 delete-authdb(1) 98
 delete-auth-realm(1) 99
 delete-custom-resource(1) 100
 delete-domain(1) 101
 delete-file-user(1) 102
 delete-http-listener(1) 103
 delete-http-qos(1) 104
 delete-iiop-cluster-endpoint(1) 105
 delete-iiop-listener(1) 107
 delete-instance(1) 108
 delete-javamail-resource(1) 110
 delete-jdbc-connection-pool(1) 111
 delete-jdbc-resource(1) 112
 delete-jmsdest(1) 113
 delete-jms-resource(1) 114
 delete-jndi-resource(1) 115
 delete-jvm-options(1) 116
 delete-lifecycle-module(1) 118
 delete-mime(1) 119
 delete-persistence-resource(1) 120
 delete-profiler(1) 121
 delete-ssl(1) 122
 delete-virtual-server(1) 123
 deploy(1) 124
 deploydir(1) 127
 disable(1) 129

display-license(1) 130
enable(1) 131
export(1) 132
flexanlg(1M) 133
get(1) 135
help(1) 136
htpasswd(1M) 141
install-license(1) 142
jms-ping(1) 143
jspc(1M) 144
list(1) 146
list-acls(1) 148
list-authdbs(1) 149
list-auth-realms(1) 150
list-components(1) 151
list-custom-resources(1) 153
list-domains(1) 154
list-file-groups(1) 155
list-file-users(1) 156
list-http-listeners(1) 157
list-iiop-cluster-config(1) 158
list-iiop-listeners(1) 159
list-instances(1) 160
list-javamail-resources(1) 161
list-jdbc-connection-pools(1) 162
list-jdbc-resources(1) 163
list-jmsdest(1) 164
list-jms-resources(1) 165
list-jndi-resources(1) 166
list-lifecycle-modules(1) 167
list-mimes(1) 168
list-persistence-resources(1) 169
list-profiler(1) 170
list-profilers(1) 171
list-sub-components(1) 172
list-virtual-servers(1) 173
multimode(1) 174
package-appclient(1M) 175

reconfig(1) 177
restart-instance(1) 179
set(1) 181
show-component-status(1) 182
show-instance-status(1) 183
shutdown(1) 184
start-appserv(1) 185
start-domain(1) 186
start-instance(1) 187
stop-appserv(1) 189
stop-domain(1) 190
stop-instance(1) 192
undeploy(1) 194
unset(1) 195
update-file-user(1) 196
verifier(1M) 197
version(1) 198
wscompile(1M) 199
wsdeploy(1M) 203

Index 207

Preface

Both novice users and those familiar with the SunOS operating system can use online man pages to obtain information about the system and its features. A man page is intended to answer concisely the question “What does it do?” The man pages in general comprise a reference manual. They are not intended to be a tutorial.

Overview

The following contains a brief description of each man page section and the information it references:

- Section 1 describes, in alphabetical order, commands available with the Sun Java System Application Server.
- Section 1M describes, in alphabetical order, the asadmin utility commands.
- Section 8 describes all the other Application Server utility commands.

Below is a generic format for man pages. The man pages of each manual section generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section.

NAME	This section gives the names of the commands or functions documented, followed by a brief description of what they do.
SYNOPSIS	<p>This section shows the syntax of commands or functions.</p> <p>The following special characters are used in this section:</p>

	<p>[] Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.</p> <p> Separator. Only one of the arguments separated by this character can be specified at a time.</p>
DESCRIPTION	This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, and functions are described under USAGE.
OPTIONS	This section lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.
OPERANDS	This section lists the command operands and describes how they affect the actions of the command.
EXAMPLES	This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command-line entry and machine response is shown. Whenever an example is given, the prompt is shown as <code>example%</code> , or if the user must be superuser, <code>example#</code> . Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS, and USAGE sections.
EXIT STATUS	This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion, and values other than zero for various error conditions.
SEE ALSO	This section lists references to other man pages, in-house documentation, and outside publications.
NOTES	This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.

BUGS

This section describes known bugs and, wherever possible, suggests workarounds.

User Commands

add-iiop-cluster-endpoint(1)

NAME	add-iiop-cluster-endpoint – adds an IIOP endpoint to the IIOP cluster.																						
SYNOPSIS	<pre> add-iiop-cluster-endpoint [--host <i>admin-host</i>] [--port <i>admin_port</i>] [--user <i>admin-username</i>] [--password <i>admin-password</i>] [--passwordfile <i>filename</i>] [--secure -s] --iiopserverinstance <i>iiop-server-instance</i> --iiopendpointhost <i>iiop-endpoint-host</i> [--iiopendpointport <i>iiop-endpoint-port</i>] [--instance <i>instance_name</i>] <i>iiop-endpoint-id</i> </pre>																						
DESCRIPTION	<p>Adds an IIOP endpoint to the IIOP cluster. For the IIOP cluster configuration changes to take effect, the instance must be reconfigured and then restarted after executing this command. The current application server instance will be added to the IIOP cluster configuration at the time of creation of the instance itself. This command creates an IIOP cluster element, and an IIOP server instance, if it does not exists already.</p> <p>The add-iiop-cluster-endpoint command is available only in the <i>Enterprise Edition</i> of the <i>Sun Java System Application Server</i>.</p>																						
OPTIONS	<table> <tr> <td>-H --host</td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td>-p --port</td><td>administrative port number associated with the administrative host.</td></tr> <tr> <td>-u --user</td><td>administrative user associated for the instance.</td></tr> <tr> <td>-w --password</td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td>--passwordfile</td><td>file containing passwords appropriate for the command (e.g., administrative instance).</td></tr> <tr> <td>-s --secure</td><td>indicates communication with the administrative instance in secured mode.</td></tr> <tr> <td>--iiopserverinstance</td><td>name of the server instance to be added to the IIOP cluster.</td></tr> <tr> <td>--iiopendpointid</td><td>identification of IIOP endpoint.</td></tr> <tr> <td>--iiopendpointhost</td><td>host name of the IIOP endpoint.</td></tr> <tr> <td>--iiopendpointport</td><td>port number of the IIOP endpoint.</td></tr> <tr> <td>--instance</td><td>name of the server instance to which this operation is targeted.</td></tr> </table>	-H --host	host name of the machine hosting the administrative instance.	-p --port	administrative port number associated with the administrative host.	-u --user	administrative user associated for the instance.	-w --password	administrative password corresponding to the administrative user.	--passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).	-s --secure	indicates communication with the administrative instance in secured mode.	--iiopserverinstance	name of the server instance to be added to the IIOP cluster.	--iiopendpointid	identification of IIOP endpoint.	--iiopendpointhost	host name of the IIOP endpoint.	--iiopendpointport	port number of the IIOP endpoint.	--instance	name of the server instance to which this operation is targeted.
-H --host	host name of the machine hosting the administrative instance.																						
-p --port	administrative port number associated with the administrative host.																						
-u --user	administrative user associated for the instance.																						
-w --password	administrative password corresponding to the administrative user.																						
--passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).																						
-s --secure	indicates communication with the administrative instance in secured mode.																						
--iiopserverinstance	name of the server instance to be added to the IIOP cluster.																						
--iiopendpointid	identification of IIOP endpoint.																						
--iiopendpointhost	host name of the IIOP endpoint.																						
--iiopendpointport	port number of the IIOP endpoint.																						
--instance	name of the server instance to which this operation is targeted.																						
OPERANDS	<i>iiop-endpoint-id</i> identification of the IIOP endpoint.																						
EXAMPLES	<p>EXAMPLE 1 Add an IIOP server instance, server1, with an endpoint, orb-listener-1, to the IIOP cluster for a given server instance, server2, with endpoint port as 3601</p> <pre> asadmin> add-iiop-cluster-config --user admin --password myPasswd --iiopserverinstance server1 --iiopendpointhost myHost --iiopendpointport 3601 --instance server2 orb-listener-1 </pre> <p>IIOP endpoint added to the IIOP cluster successfully.</p>																						

add-iiop-cluster-endpoint(1)

EXAMPLE 2 Add an IIOP server instance, server1, with an endpoint, orb-listener-1, to the IIOP cluster for a given server instance, server2, with default endpoint port, 3600

```
asadmin> add-iiop-cluster-config --user admin --password myPasswd  
--iiopserverinstance server1 --iiopendpointid endpoint1 --iiopendpointhost myHost  
3700 server2
```

IIOP endpoint added to the IIOP cluster successfully.

SEE ALSO delete-iiop-cluster-endpoint(1), list-iiop-cluster-config(1)

add-resources(1)

NAME	add-resources – registers the named resource in the XML file specified.
SYNOPSIS	add-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>xml_file_path</i>
DESCRIPTION	Registers the named resource in the XML file specified. The <i>xml_file_path</i> 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	<i>xml_file_path</i> path to the XML file containing the resource(s) to be registered.
EXAMPLES	<p>EXAMPLE 1 Using add-resources</p> <pre>asadmin> add-resources --user admin --passwordfile passwords.txt --host localhost --port 4848 --instance server1 resource.xml</pre> <p>Created the resource</p> <p>Where: resource.xml is the resource file containing resources to be created.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-jdbc-connection-pool(1), list-jdbc-resource(1), create-jms-resource(1), create-jndi-resource(1), create-javamail-resource(1), create-persistence-resource(1), create-custom-resource(1)

NAME	appclient – launches the Application Client Container and invokes the client application packaged in the application JAR file												
SYNOPSIS	appclient -client <i>client_application_jar</i> [-mainclass <i>client_application_main_classname</i> -name <i>display_name</i>] [-xml <i>sun-acc.xml file</i>] [-textauth] [app-args]												
DESCRIPTION	<p>Use the appclient command to launch the application client container and invoke a client application that is packaged in an application JAR file.</p> <p>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.</p>												
OPTIONS	<table> <tr> <td>-client</td><td>required; the name and location for the client application jar file.</td></tr> <tr> <td>-mainclass</td><td>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.</td></tr> <tr> <td>-name</td><td>optional; the display name for the client application. Used for multiple client applications.</td></tr> <tr> <td>-xml</td><td>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.</td></tr> <tr> <td>-textauth</td><td>optional; used to specify using text format authentication when authentication is needed.</td></tr> <tr> <td>app-args</td><td>optional; represents a list of arguments, separated by spaces, passed to the clients main() method.</td></tr> </table>	-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.
-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	<p>EXAMPLE 1 Using the appclient command</p> <pre>appclient -client sunoneappserv/bin/myclientapp.jar -mainclass com.sun.test.TestAppClient -xml sun-acc.xml scott sample</pre> <p>Where: <i>sunoneappserv/bin/myclientapp.jar</i> is the full path for the client application .jar file, <i>com.sun.test.TestAppClient</i> is the full Java package name of the main client application, <i>scott</i> and <i>sample</i> are arguments to pass to the application, and <i>sun-acc.xml</i> is the name of the client configuration XML file. If <i>sun-acc.xml</i> 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.</p>												
ATTRIBUTES	See attributes(5) for descriptions of the following attributes:												

appclient(1M)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO package-appclient(1M), asadmin(1M)

NAME	asadmin – utility for performing administrative tasks for the Sun Java System Application Server
SYNOPSIS	asadmin <i>subcommand</i> [-short_option [<i>short_option_argument</i>]] * [--long_option [<i>long_option_argument</i>]] * [<i>operand</i>] *
DESCRIPTION	<p>Use the asadmin utility to perform any administrative task for the Sun Java System Application Server. You can use this utility in place of using the Administrator interface.</p> <p>The <i>subcommand</i> 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.</p> <p>asadmin can be used in command shell invocation or multi command mode (known as <i>multimode</i>). 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 <i>multimode</i>. You may provide commands by passing a previously prepared list of commands from a file or standard input (pipe). Additionally, you can invoke <i>multimode</i> from within a <i>multimode</i> session; once you exit the second <i>multimode</i> environment, you return to your original <i>multimode</i> environment.</p> <p>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 <i>multimode</i> when you run one subcommand at a time from the command prompt; and when you run in <i>multimode</i> from a file. Subcommands in <i>multimode</i>, when piped from an input stream, and subcommands invoked from another program, cannot run in the interactive option.</p> <p>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.</p> <p>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.</p>

asadmin(1M)

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 Java System 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.

Environment Subcommands

See the *Sun Java System Application Server Administration 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).

```
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.

```
multimode [--file filename] [--encoding encode] [--passwordfile filename] [--interactive]
```

- 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.

Domain Administration Subcommands

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
```

asadmin(1M)

Instance Subcommands

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.

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

- 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.

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 instance_port instance_name

- The named instance must not exist within that domain.

start-instance [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--domain domain_name] [--debug=false] [--passwordfile filename] [--secure|-s] instance_name

- 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.

```
delete-instance [--user admin_user] [--password admin_password] [--host localhost]
[--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename]
[--secure|-s] instance_name
```

- 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.

```
stop-instance [--user admin_user] [--password admin_password] [--host localhost]
[--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename]
[--secure|-s] instance_name
```

- 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.

```
restart-instance [--user admin_user] [--password admin_password] [--host localhost]
[--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename]
[--secure|-s] instance_name
```

- 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] [--domain domain_name] [--local=false] [--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.

List and Status Subcommands

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 Java System Application Server.

```
deploy --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--virtualservers virtual_servers]
[--type application|ejb|web|connector] [--contextroot context_root] [--force=true]
[--precompilejsp=false] [--verify=false] [--name component_name]
[--upload=true] [--retrieve local_dirpath] [--instance instance_name]
filepath
```

- 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.

```
deploydir --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--virtualservers virtual_servers]
[--type application|ejb|web|connector] [--contextroot context_root] [--force=true]
[--precompilejsp=false] [--verify=false] [--name component_name]
[--instance instance_name] dirpath
```

- 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*.

```
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
```

- Removes the component from the named instance.

Configuration Subcommands

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

```
get [--monitor] --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] attributename [attribute_name] *
```

- 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 Java System Application Server Administration Guide* for a listing of the valid attribute names.

```
set [--monitor] --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] attributename=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.

```
reconfig --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s]
[--discardmanualchanges=false] [--keepmanualchanges=false] instance_name
```

- 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] [--secure|-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.

iMQ Administration Subcommands

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

```
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
```

- 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.

Resource Administration Subcommands

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

```
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] [--isolationguaranteed=true]
[--isconnectvalidatereq=false] [--validationmethod auto-commit]
[--validationtable table_name] [--failconnection=false] [--description text]
[--property (name=value):[name=value] *] connection_pool_ID
```

- `--datasourceclassname` is the 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 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.
- `--isolationguaranteed` 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.

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

- Removes the JDBC connection pool from the named instance.

```
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.

```
create-jdbc-resource --user admin_user[--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--connectionpoolid ID [--enabled=true] [--description text] jndi_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.

- `--resourcetype` is the JMS resource type which can be: `javax.jms.Topic`, `javax.jms.Queue`, `javax.jms.TopicConnectionFactory`, `javax.jms.QueueConnectionFactory`.

```
create-jndi-resource --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--jndilookupname lookup_name --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]
[--transportprotocol=smtp] [--transportprotocolclass=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]
[--transportprotocol=smtp] [--transportprotocolclass=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
```

- 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] [--secure|-s] [--instance instance_name]
jndi_name
```

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

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

- Gets all the custom resources from the specified instance.

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

- Registers the named resource in the XML file identified.
- 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]
[--port 4848] [--passwordfile filename] [--secure|-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]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
listener_ID
```

- Removes the custom resource with the given IIOp listener from the specified instance.

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

- 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]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
```

asadmin(1M)

MIME Subcommands

```
--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.

```
delete-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.

```
list-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.

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

- --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.

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

- 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 instance_name] mime_ID
```

- Gets all the MIMEs from the specified instance.

HTTP Listener Subcommands

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

```
create-http-listener --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] --address address
[--instance instance_name] --listenerport listener_port --defaultvs virtual_server
--servername server_name [--family family] [--acceptorthreads acceptor_threads]
[--blockingenabled blocking_enabled] [--securityenabled security_enabled]
[--enabled=enabled] listener_ID
```

- `--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 list on port 443 is recommended.
- `--defaultvs` 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.

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

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

```
list-http-listeners --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
httplistener_ID
```

- Gets all the HTTP listeners from the specified instance.

HTTP QOS Subcommands

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

```
create-http-qos --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--virtualserver virtual_server_ID]
[--bwlimit bwlimit] [--enforcebwlimit=enforce_bw_limit] [--connlimit connection_limit]
[--enforceconnlimit=enforce_conn_limit] instance_name
```

asadmin(1M)

Authorization Database Subcommands

- `--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.
- `--connnlimit` is the maximum number of concurrent connections for the server, virtual server class, or virtual server.
- `--enforceconnnlimit` determines whether the connection limit should be enforced or not.

```
delete-http-qos --user admin_user [--password admin_password]
[--host localhost] [--port 4848] [--passwordfile filename] [--secure|-s]
[--virtualserver virtual_server_ID] instance_name
```

- 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.

```
create-authdb --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--database database--virtualserver virtualserver_ID [--basedn basedn]
[--certmaps certmaps] authdb_ID
```

- `--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.

```
delete-authdb --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--virtualserver virtualserver_ID authdb_ID
```

- `--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.

```
list-authdbs --user admin_user [--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--virtualserver virtualserver_ID authdb_ID
```


Authorization Realm Subcommands

- `--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|-s] [--instance instance_name]
--classname realm_class[--property (name=value) [: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|-s] [--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|-s] instance_name
```

- Gets all the authorization realms from the specified instance.

ACL Subcommands

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

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

- 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.

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

- 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|-s] instance_name
```

- Gets all the ACLs from the specified instance.

Virtual Server Subcommands

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.

```
create-virtual-server --user admin_user[--password admin_password] [--host localhost]
[--port 4848] [--passwordfile filename] [--secure|-s] [--instance instance_name]
--hosts hosts--mime mime_types_file[--httplisteners http_listeners]
[--defaultwebmodule default_web_module] [--configfile config_file] [--defaultobj default_object]
```

asadmin(1M)

```
[--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 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.

Profiler Subcommands

```
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
```

- --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.

```
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] [--ssl3tlsciphers ssl3_tls_ciphers]
[--tlsenabled=true] [--tlsrollbackenabled=true] [--clientauthenenabled=false]
[listener_id]
```

- `--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 TLS 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.
- `--clientauthenenabled` 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]
```

JVM Options Subcommands

- `--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_name] *
```

- 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.

License Subcommands

```
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] [--passwordfile filename] [--secure|-s]
```

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

File User Subcommands

```
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_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]
```

**System
Subcommands**

```
[--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 Java System 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:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO

`appclient(1AS)`, `package-appclient(1AS)`

asant(1M)

NAME	asant – launches the Jakarta Ant tool																						
SYNOPSIS	asant <i>target_list</i>																						
DESCRIPTION	<p>Use the <code>asant</code> command to automate repetitive development and deployment tasks. <code>asant</code> is a shell script that invokes the underlying Ant infrastructure after initializing the environment to pickup the application server installed targets.</p> <p>To use Ant as part of the Sun Java System Application Server, verify that your PATH includes the provided <code>asant</code> (Solaris) <code>ant.bat</code> (Windows) script.</p> <p>The bundled sample applications use <code>asant</code> extensively; however, <code>asant</code> can be used in any development or operational environments.</p> <p>The build targets are represented in the <code>build.xml</code> files that accompany the sample applications.</p> <p>To use the Ant tool to compile and reassemble the sample applications, verify that the <code>\$AS_INSTALL/bin</code> directory is on your environment's path. On UNIX, add the <code>\$AS_INSTALL/bin</code> directory to your PATH environment variable. On Windows, after installing the Sun Java System Application Server, set the system path by adding <code>\$AS_INSTALL\bin</code> 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.</p> <p>The <i>target_list</i> is one or more space separated tasks as described below.</p>																						
TARGETS	<table><tr><td><code>compile</code></td><td>compiles all Java source code.</td></tr><tr><td><code>jar</code></td><td>assembles the EJB JAR module.</td></tr><tr><td><code>war</code></td><td>assembles the WAR file in <code><sample_dir>/assemble/war</code></td></tr><tr><td><code>ear</code></td><td>assembles the EAR file in <code><sample_dir>/assemble/ear</code></td></tr><tr><td><code>core</code></td><td>(default) compiles all sources, builds stubs and skeletons; and assembles EJB JAR, WAR and EAR files. This is the default target for all <code>build.xml</code> files shipped in the Sun Java System Application Server.</td></tr><tr><td><code>javadocs</code></td><td>creates Java docs in <code><sample_dir>/javadocs</code></td></tr><tr><td><code>all</code></td><td>builds core and javadocs , verifies and deploys the application, and adds the resources..</td></tr><tr><td><code>deploy</code></td><td>deploys the application and automatically expands the EJB JAR; does not install Javadocs.</td></tr><tr><td><code>undeploy</code></td><td>removes the deployed sample from the Sun Java System Application Server.</td></tr><tr><td><code>clean</code></td><td>removes <code><appname>/build/</code> and <code><appname>/assemble/</code> and <code><appname>/javadocs</code> directories.</td></tr><tr><td><code>verify</code></td><td>verifies the deployment descriptors in the sample.</td></tr></table>	<code>compile</code>	compiles all Java source code.	<code>jar</code>	assembles the EJB JAR module.	<code>war</code>	assembles the WAR file in <code><sample_dir>/assemble/war</code>	<code>ear</code>	assembles the EAR file in <code><sample_dir>/assemble/ear</code>	<code>core</code>	(default) compiles all sources, builds stubs and skeletons; and assembles EJB JAR, WAR and EAR files. This is the default target for all <code>build.xml</code> files shipped in the Sun Java System Application Server.	<code>javadocs</code>	creates Java docs in <code><sample_dir>/javadocs</code>	<code>all</code>	builds core and javadocs , verifies and deploys the application, and adds the resources..	<code>deploy</code>	deploys the application and automatically expands the EJB JAR; does not install Javadocs.	<code>undeploy</code>	removes the deployed sample from the Sun Java System Application Server.	<code>clean</code>	removes <code><appname>/build/</code> and <code><appname>/assemble/</code> and <code><appname>/javadocs</code> directories.	<code>verify</code>	verifies the deployment descriptors in the sample.
<code>compile</code>	compiles all Java source code.																						
<code>jar</code>	assembles the EJB JAR module.																						
<code>war</code>	assembles the WAR file in <code><sample_dir>/assemble/war</code>																						
<code>ear</code>	assembles the EAR file in <code><sample_dir>/assemble/ear</code>																						
<code>core</code>	(default) compiles all sources, builds stubs and skeletons; and assembles EJB JAR, WAR and EAR files. This is the default target for all <code>build.xml</code> files shipped in the Sun Java System Application Server.																						
<code>javadocs</code>	creates Java docs in <code><sample_dir>/javadocs</code>																						
<code>all</code>	builds core and javadocs , verifies and deploys the application, and adds the resources..																						
<code>deploy</code>	deploys the application and automatically expands the EJB JAR; does not install Javadocs.																						
<code>undeploy</code>	removes the deployed sample from the Sun Java System Application Server.																						
<code>clean</code>	removes <code><appname>/build/</code> and <code><appname>/assemble/</code> and <code><appname>/javadocs</code> directories.																						
<code>verify</code>	verifies the deployment descriptors in the sample.																						

EXAMPLES**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

Apache Software Foundation at <http://www.apache.org>, Jakarta Ant documentation at <http://jakarta.apache.org/ant/index.html>.

SUNWant documentation located in `/usr/sfw/share/doc/ant`

, `asadmin(1M)`

capture-schema(1M)

NAME	capture-schema – stores the database metadata (schema) in a file for use in mapping and execution
SYNOPSIS	capture-schema -dburl <i>url</i> -username <i>name</i> -password <i>password</i> -driver <i>a_jdbc_driver</i> [-schemaname <i>name</i>] [-table <i>tablename</i>] * [-out <i>filename</i>]
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	<p>-dburl JDBC URL expected by the driver for accessing a database.</p> <p>-username user name for authenticating access to a database.</p> <p>-password password for accessing the selected database.</p> <p>-driver JDBC driver classname in your CLASSPATH.</p> <p>-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.</p> <p>-table name of the table; multiple table names can be specified.</p> <p>-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.</p>
EXAMPLES	<p>EXAMPLE 1 Using capture-schema</p> <pre>capture-schema -dburl jdbc:oracle:thin:@sadbtrue:1521:ora817 -schemaname cantiflas -username cantiflas -password enigma -driver oracle.jdbc.driver.OracleDriver</pre>
SEE ALSO	asadmin(1M)

NAME	cladmin – runs certain asadmin commands simultaneously on all application server instances in a cluster																						
SYNOPSIS	cladmin [--help] [--instancefile <i>instance_file_location</i>] [--passwordfile <i>password_file_location</i>] <i>asadmin_command</i>																						
DESCRIPTION	<p>To simplify the task of cluster administration, you can use the <code>cladmin</code> command to run the following <code>asadmin</code> commands simultaneously on all application server instances in a cluster:</p> <table> <tr> <td><code>start-instance</code></td><td>starts a server instance and all the services associated with it</td></tr> <tr> <td><code>stop-instance</code></td><td>stops the specified server instance and all the services associated with it</td></tr> <tr> <td><code>deploy</code></td><td>deploys the specified component</td></tr> <tr> <td><code>undeploy</code></td><td>removes the component from the named instance</td></tr> <tr> <td><code>create-jdbc-resource</code></td><td>registers the JDBC resource to the named instance</td></tr> <tr> <td><code>create-jdbc-connection-pool</code></td><td>registers the JDBC connection pool to the named instance</td></tr> <tr> <td><code>configure-session-persistence</code></td><td>enables configuration of parameters related to session persistence</td></tr> <tr> <td><code>delete-jdbc-resource</code></td><td>removes the JDBC resource from the named instance</td></tr> <tr> <td><code>delete-jdbc-connection-pool</code></td><td>removes the JDBC connection pool from the named instance</td></tr> </table> <p>To ensure consistency in configuration of all application server instances in a cluster, use the <code>cladmin</code> command for all the supported <code>asadmin</code> commands.</p> <p>The following input files are required for the <code>cladmin</code> command to function:</p> <table> <tr> <td><code>clinstance.conf</code></td><td>contains information about the application server instances that are part of the cluster.</td></tr> <tr> <td><code>clpassword.conf</code></td><td>contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.</td></tr> </table> <p>By default the input files are located in the application server configuration directory which is located in <code>/etc/opt/SUNWappserver7</code>. You can modify the input files to support different configurations. The server instances that are part of the cluster must be defined in the instance file (default name is <code>clinstance.conf</code>).</p> <p>Before editing the input files, keep in mind:</p> <ul style="list-style-type: none"> ■ The order of the entries must not be changed. ■ Any line that starts with a hash mark (#) is treated as a comment. 	<code>start-instance</code>	starts a server instance and all the services associated with it	<code>stop-instance</code>	stops the specified server instance and all the services associated with it	<code>deploy</code>	deploys the specified component	<code>undeploy</code>	removes the component from the named instance	<code>create-jdbc-resource</code>	registers the JDBC resource to the named instance	<code>create-jdbc-connection-pool</code>	registers the JDBC connection pool to the named instance	<code>configure-session-persistence</code>	enables configuration of parameters related to session persistence	<code>delete-jdbc-resource</code>	removes the JDBC resource from the named instance	<code>delete-jdbc-connection-pool</code>	removes the JDBC connection pool from the named instance	<code>clinstance.conf</code>	contains information about the application server instances that are part of the cluster.	<code>clpassword.conf</code>	contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.
<code>start-instance</code>	starts a server instance and all the services associated with it																						
<code>stop-instance</code>	stops the specified server instance and all the services associated with it																						
<code>deploy</code>	deploys the specified component																						
<code>undeploy</code>	removes the component from the named instance																						
<code>create-jdbc-resource</code>	registers the JDBC resource to the named instance																						
<code>create-jdbc-connection-pool</code>	registers the JDBC connection pool to the named instance																						
<code>configure-session-persistence</code>	enables configuration of parameters related to session persistence																						
<code>delete-jdbc-resource</code>	removes the JDBC resource from the named instance																						
<code>delete-jdbc-connection-pool</code>	removes the JDBC connection pool from the named instance																						
<code>clinstance.conf</code>	contains information about the application server instances that are part of the cluster.																						
<code>clpassword.conf</code>	contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.																						

cladmin(1M)

Before running the `cladmin` command you must:

- have the `asadmin` command on the local machine.
- use the `clsetup` command to setup a typical cluster configuration.
- use the same administrator password for all domains that are part of the cluster.
- start the administration servers of all instances that are part of the cluster.
- make sure you list all the instances that are part of the cluster in the `clinstance.conf` file.
- change the directory to `install_dir/bin`; where `install_dir` is the server installation directory.

During standard installation, the `clinstance.conf` file is created with entries for two instances. If you add more instances to the cluster, or create more clusters, you must update the `clinstance.conf` file with the information for the new instances. The information about the instances in each cluster must be identified in the instance file (default name is `clinstance.conf`). The following are the possible entries in the `clinstance.conf` file:

TABLE 1 Entries in the `clinstance.conf` file

Entry	Definition	Default Value(s)
instance	name of the application server instance	server1, server2
user	Administration server user name	admin
host	Host name	localhost
port	Port number of the Administration Server	4848
domain	Name of the Administrative domain	domain1
instanceport	Port number of the application server instance	80, 81

The `clpassword.conf` file contains the administration server password. During execution of the `cladmin` command, the `asadmin` command requires the administration server password specified in the `clpassword.conf` file. The format for the `clpassword.conf` file is :

`AS_ADMIN_PASSWORD=password`

Where *password* is the administration server password.

Permissions 0600 are preset on the `clpassword.conf` file and it can be accessed only by the root user.

A log file, named `cladmin.log`, is available in the `/var/tmp/cladmin.log` directory. By default the `cladmin` command executes in verbose mode and logs information in the log file. Log file entries start and end with timestamp tags. If the log file exists prior to execution, the output is appended to the existing log file. Scan the log after each execution to verify that it ran properly.

If you are running multiple clusters, a separate `clinstance.conf` file must be specified for each cluster. You can run the `cladmin` command on the instances in any cluster by specifying the path of the input files for that particular cluster.

OPTIONS	<code>--help</code>	displays the syntax of the command
	<code>--instancefile</code>	filepath location to the instance file (default is <code>clinstance.conf</code>).
	<code>--passwordfile</code>	filepath location to the password file (default is <code>clpassword.conf</code>).

EXAMPLES **EXAMPLE 1** Using `cladmin` to start all instances in a cluster

```
./cladmin start-instance
```

Since the location of the input files is not specified, the input files are read from the configuration directory (located by default in `/etc/opt/SUNWappserver7`).

EXAMPLE 2 Using `cladmin` to start all instances with Input file locations specified

```
./cladmin --instancefile /tmp/clinstance.conf
--passwordfile /tmp/clpassword.conf start-instance
```

EXAMPLE 3 Using `cladmin` to create a JDBC connection pool

```
./cladmin create-jdbc-connection-pool --user admin
--datasourceclassname com.sun.hadb.jdbc.ds.HadbDataSource
--isolationguaranteed=true --isolationlevel repeatable-read
--isconnectvalidatereq=true --isconnectvalidatereq=true
--validationmethod auto-commit --failconnection=false --property
username=test:password=test:serverList=exampleserver1.example.
com\:15100,exampleserver2.example.com\:15120 CluJDBC
```

EXAMPLE 4 Using `cladmin` to configure session persistence for all instances in a cluster

```
./cladmin configure-session-persistence --user admin
--type ha --frequency web-method --scope session --store jdbc_hastore
```

EXIT STATUS	0	successful exit
	2	syntax error

cladmin(1M)

- 3 instance file not found
- 4 instance file cannot be read
- 5 password file not found
- 6 password file cannot be read
- 7 user trying to run the command is not a root user
- 8 cannot locate `asadmin` command
- 9 cannot create temporary file
- 10 command could not be executed

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO `asadmin(1M)`, `hadbm(1M)`, `clsetup(1M)`

NAME	clear-session-store – clears all the sessions from the persistent store								
SYNOPSIS	<pre>clear-session-store [--storeurl <i>persistent_store_URL</i>] [--storeuser <i>username</i>] [--storepassword <i>user_password</i>] [--optionsfile <i>filename</i>]</pre>								
DESCRIPTION	<p>Removes all the sessions, including passivated sessions, from the persistent store based on:</p> <table> <tr> <td>storeurl</td><td>The JDBC URL that is pointing to the persistent store being used for saving the session data.</td></tr> <tr> <td>storeuser</td><td>The username associated with the persistent store being used for saving session data.</td></tr> <tr> <td>storepassword</td><td>The password corresponding to the user of the persistent store being used for saving session data.</td></tr> </table> <p>Use this command with caution.</p> <p>All the options should be specified either on the command-line itself, or in an options file. If a password is not specified on the command-line, you will be prompted to provide it, if you are in the interactive mode.</p> <p>This command can only be run locally.</p> <p>The <code>clear-session-store</code> command is available only in the <i>Enterprise Edition</i> of the Sun Java System Application Server.</p>	storeurl	The JDBC URL that is pointing to the persistent store being used for saving the session data.	storeuser	The username associated with the persistent store being used for saving session data.	storepassword	The password corresponding to the user of the persistent store being used for saving session data.		
storeurl	The JDBC URL that is pointing to the persistent store being used for saving the session data.								
storeuser	The username associated with the persistent store being used for saving session data.								
storepassword	The password corresponding to the user of the persistent store being used for saving session data.								
OPTIONS	<table> <tr> <td>--storeurl</td><td>JDBC URL of the persistent store.</td></tr> <tr> <td>--storeuser</td><td>username associated for the persistent store.</td></tr> <tr> <td>--storepassword</td><td>password corresponding to the storeuser.</td></tr> <tr> <td>--optionsfile</td><td>a valid filename that has the options in it.</td></tr> </table>	--storeurl	JDBC URL of the persistent store.	--storeuser	username associated for the persistent store.	--storepassword	password corresponding to the storeuser.	--optionsfile	a valid filename that has the options in it.
--storeurl	JDBC URL of the persistent store.								
--storeuser	username associated for the persistent store.								
--storepassword	password corresponding to the storeuser.								
--optionsfile	a valid filename that has the options in it.								
EXAMPLES	<p>EXAMPLE 1 Using clear-session-store to clear all sessions</p> <pre>asadmin> clear-session-store --storeurl jdbc:sun:hadb@myhost1:7676,myhost2:6767 --storeuser haadmin --storepassword hapasswd</pre>								
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command				
0	command executed successfully								
1	error in executing the command								
SEE ALSO	configure-session-persistence(1), create-session-store(1)								

clsetup(1M)

NAME	clsetup – automates the cluster setup process						
SYNOPSIS	<pre>clsetup [--help] [--instancefile <i>instance_file</i>] [--resourcefile <i>resource_file</i>] [--passwordfile <i>password_file</i>]</pre> <pre>clsetup [--help] [--instancefile <i>instance_file</i>] [--passwordfile <i>password_file</i>] <i>verify</i></pre>						
DESCRIPTION	<p>Use <code>clsetup</code> command to simplify the task of setting up a cluster on a typical machine configuration. The <code>clsetup</code> command can create instances if they do not already exist, and perform cluster setup on these instances based on the configurations in the input files.</p> <p>The following input files are required for the <code>clsetup</code> command to function:</p> <table><tr><td><code>clinstance.conf</code></td><td>contains information about the application server instances that are part of the cluster.</td></tr><tr><td><code>clresource.conf</code></td><td>contains information about the HADB database and other resource information.</td></tr><tr><td><code>clpassword.conf</code></td><td>contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.</td></tr></table> <p>By default the input files are located in the application server configuration directory which is located in <code>/etc/opt/SUNWappserver7</code>. Additionally, the <code>clsetup</code> command is located by default in <code>installdir/bin</code>.</p> <p>Before editing the input files, keep in mind:</p> <ul style="list-style-type: none">■ The order of the entries must not be changed.■ Any line that starts with a hash mark (#) is treated as a comment.■ If the entities to be handled (HADB database nodes and application server instances) already exist, the <code>clsetup</code> command does not delete or reconfigure them, and the respective configuration steps are skipped.■ <code>clsetup</code> command creates the HADB database with no <code>inetd</code> settings. Consequently, <code>inetd</code> configurations are not performed.■ <code>clsetup</code> command does not set any shared memory values. <p>Before running the <code>clsetup</code> command you must:</p> <ul style="list-style-type: none">■ have the <code>asadmin</code> and <code>hadbm</code> command on the local machine.■ use the <code>clsetup</code> command as-is to setup a typical cluster configuration.■ run the HADB shared memory setup, and HADB cluster host communication setup for SSH and RSH.■ use the same administrator password for all domains that are part of the cluster.■ start the administration servers of all instances that are part of the cluster.	<code>clinstance.conf</code>	contains information about the application server instances that are part of the cluster.	<code>clresource.conf</code>	contains information about the HADB database and other resource information.	<code>clpassword.conf</code>	contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.
<code>clinstance.conf</code>	contains information about the application server instances that are part of the cluster.						
<code>clresource.conf</code>	contains information about the HADB database and other resource information.						
<code>clpassword.conf</code>	contains the <code>asadmin</code> password and is pre-populated with the correct password during a standard installation.						

During standard installation, the `clinstance.conf` file is created with entries for two instances. This occurs only if the Sun Java System Application Server is also installed. However, if only the application server administration client is installed on the machine, the `clinstance` and `clpassword` files installed on this machine are not pre-populated. If you add more instances to the cluster, you must update the `clinstance.conf` file with the information for the new instances. The information about the instances in each cluster must be defined in the instance file (default name is `clinstance.conf`). The following are the possible entries in the `clinstance.conf` file:

TABLE 1 Entries in the `clinstance.conf` file

Entry	Definition	Default Value(s)
instance	Name of the application server instance	server1, server2
user	Administration server user name	admin
host	Host name	localhost
port	Port number of the Administration Server	4848
domain	Name of the Administrative domain	domain1
instanceport	Port number of the application server instance	80, 81
master	Boolean to indicate the master instance	true. For first time installation.

The `clresource.conf` file contains information about the HADB database, JDBC connection pool, JDBC resource, session store, and session persistence. Permissions 0600 are preset on the `clresource.conf` file and it can be accessed only by the root user. The following are the possible entries in the `clresource.conf` file:

TABLE 2 Entries in the `clresource.conf` file

Entry	Definition	Default Value
HADB Database Information		
historypath	Path for the history files	/var/tmp
devicepath	Path for the data and log devices	installdir/SUNWhadb/4
datadevices	The number of data devices on each node	1

clsetup(1M)

TABLE 2 Entries in the clresource.conf file (Continued)

Entry	Definition	Default Value
portbase	The port base number used for node 0	15200
spares	The number of spare nodes	0
set	Comma separated list of database configuration attributes.	managementProtocol=rsh To specify the use of RSH instead of SSH (the default), uncomment the following line:#set managementProtocol=rsh
inetd	Database runs with inetd daemon	false
inetdsetupdir	The inetd setup files directory location	/tmp
devicesize	Size of the device in MB	512
dbpassword	Password for the system user of the database	password
hosts	All hosts used for all data nodes	localhost,localhost
JDBC Connection Pool Information		
steadypoolsize	Minimum and initial number of connections maintained in the pool	8
maxpoolsize	Maximum number of connections that can be created	32
datasourceclassname	Name of the vendor supplied JDBC datasource	com.sun.hadb.jdbc.ds.HadbDataSource
isolationlevel	Transaction isolation level on the pooled database connections	repeatable-read
isolationguaranteed	Transaction isolation level guaranteed	true
validationmethod	type of validation method	meta-data
property	Property used to specify username, password, and resource configuration	username=appservusr:password=password:cacheDataBaseMetaData=false:eliminateRedundantEndTransaction=true:serverList=replaceurl
JDBC Resource Information		

TABLE 2 Entries in the clresource.conf file (Continued)

Entry	Definition	Default Value
connectionpoolid	Name of the connection pool	appservCPL
Session Store Information		
storeurl	JDBC URL of the HADB database	<i>replaceurl</i> Value is replaced by the actual URL at runtime.
storeuser	HADB user who has access to session store	appservusr
storepassword	password for store user	password
dbssystempassword	Database system password	password
Session Persistence Information		
type	The session persistence type	ha
frequency	The session persistence frequency	web-method
scope	The session scope	session
store	Session store	jdbc/hastore
Stateful Session Bean Information		
sfsb	The stateful session bean failover	true
RMI IIOP Failover Information		
rmi_iiop	RMI IIOP cluster configuration	true
Cluster ID Information		
cluster_id	Cluster ID	cluster1

The clpassword.conf file contains the administration server password. During execution of the cladmin command, the asadmin command requires the administration server password specified in the clpassword.conf file. The format for the clpassword.conf file is :

AS_ADMIN_PASSWORD=*password*

Where *password* is the administration server password.

clsetup(1M)

	Permissions 0600 are preset on the <code>clpassword.conf</code> file and it can be accessed only by the root user.	
	A log file, named <code>clsetup.log</code> , is available in the <code>/var/tmp</code> directory. By default the <code>clsetup</code> command executes in verbose mode and logs information in the log file. Log file entries start and end with timestamp tags. If the log file exists prior to execution, the output is appended to the existing log file. Scan the log after each execution to verify that it ran properly.	
OPTIONS	<code>--help</code>	displays the syntax of the command
	<code>--instancefile</code>	filepath location to the instance filename (default is <code>install_config_dir/ clinstance.conf</code>).
	<code>--resourcefile</code>	filepath location to the resource filename (default is <code>install_config_dir/ clresource.conf</code>).
	<code>--passwordfile</code>	filepath location to the password filename (default is <code>install_config_dir/ clpassword.conf</code>).
OPERANDS	<i>verify</i>	Compares the configuration of the master instance with the other instances in the cluster. The applications, resources and cluster configuration is compared and the differences are written to the <code>clsetup.log</code> file. The <code>--verify</code> option cannot be used if <code>--resourcefile</code> is specified.
EXIT STATUS	0	successful exit
	2	syntax error
	3	instance file not found
	4	instance file cannot be read
	5	resource file not found
	6	resource file cannot be read
	7	password file not found
	8	password file cannot be read
	10	script cannot find <code>asadmin</code>
	11	script cannot find <code>hadbm</code>
	12	cannot create temporary file
	13	session store configuration failed
	14	create HADB database failed
	15	HADB <code>get JDBC</code> URL failed
	16	user exits in welcome message
	17	cluster verification failed

clsetup(1M)

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO asadmin(1M), cladmin(1M), hadbm(1M)

configure-session-persistence(1)

NAME	configure-session-persistence – enables configuration of parameters related to session persistence
SYNOPSIS	<pre> configure-session-persistence --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--type <i>persistence_type</i>] [--frequency <i>frequency</i>] [--scope <i>scope</i>] [--store <i>jdbc_resource_jndi_name</i>] [--property <i>name=value</i> [:<i>name=value</i>]*] <i>instance_name</i> </pre>
DESCRIPTION	<p>Configure the session persistence to balance your needs for performance, reliability, and high availability.</p> <p>The <code>configure-session-persistence</code> command is available only in the <i>Enterprise Edition</i> of the Sun Java System Application Server.</p> <ul style="list-style-type: none"> Set the persistence <code>type</code> to define the location where session data is stored. The persistence types available are: <ul style="list-style-type: none"> <code>memory</code> if session persistence for the application server instance is disabled, then <code>memory</code> is the default persistence type. <code>memory</code> persistence type provides no session persistence in a clustered environment. The <code>memory</code> persistence type is intended for development environments and not to be used for production. <code>file</code> provides no session persistence in a clustered environment. Use <code>file</code> persistence type to store session data in a file. If the instance becomes unavailable and restarts, it can recover the session information that was last written to the files. The <code>file</code> persistence type is meant for development environments and not to be used for production. <code>ha</code> allows you to store session data in the HADB. The <code>ha</code> persistence type enables failover of session information between application server instances in a cluster. The session information for each application server instance in a cluster is stored in the HADB. The session information is available to all other instances in the cluster. If an instance in a cluster becomes unavailable, another instance in the cluster can continue to serve the sessions that the now unavailable instance was serving. Set the persistence <code>frequency</code> to define the frequency at which the session state is stored in the HADB. The persistence frequencies available are: <ul style="list-style-type: none"> <code>web-method</code> the session is stored after every web request. Use this frequency when you need very high availability of updated session states. <code>time-based</code> the session is stored at time intervals defined in the <code>reapIntervalSeconds</code> property of the manager property. A better throughput is achieved because the session is stored after a configurable time interval instead of after every web request. Set the persistence <code>scope</code> to define how much of the session will be saved. The persistence scope available are:

configure-session-persistence(1)

OPTIONS	modified-session	the session is saved only if it has been modified. If the persistence frequency is web-method, the entire session is stored at the end of every web request just before sending a response back to the client only if the session has changed from the last time it was stored. If the persistence frequency is time-based, the entire session is stored after each time-based frequency you specify only if the session has changed from the last time it was stored.
	session	the entire session is saved every time session information is saved to the HADB. If the persistence frequency is web-method, the entire session is stored at the end of every web request, just before sending a response back to the client. If the persistence is time-based, the entire session is stored after each time-based frequency you specify.
	modified-attributes	only the modified attributes of the session are saved. Using this mode can improve the throughput and response time significantly for applications for which only a small portion of the session state is modified for any given request.
	<ul style="list-style-type: none"> ■ Configure other session persistence properties to fine-tune the session persistence configuration. Modify the properties of manager-properties and of store-properties subelements of the session-manager element in the server.xml file (for instance-level configuration) or in the sun-web.xml file (for application-level configuration). ■ Specify the JNDI name of the JDBC resource for the HADB. If you use the HADB as the persistence store, this information is used to connect to the HADB. ■ Specify the name of the cluster to which the application server instance belongs. 	
	-u --user	administrative user associated for the instance.
	-w --password	administrative password corresponding to the administrative user.
	-H --host	host name of the machine hosting the administrative instance.
	-p --port	administrative port number associated with the administrative host.
	--type	type of store to be used for session data.
	--frequency	frequency at which session data should be saved.
OPERANDS	--scope	indicates the portion of session data that is to be saved.
	--store	JNDI name of the persistence store JDBC resource.
	--property	name/value pairs used for specifying session persistence specific attributes to customize the session persistence runtime.
	instance_name	name of the for which the session persistence has to be configured.

configure-session-persistence(1)

EXAMPLES

EXAMPLE 1 Using configure-session-persistence

```
asadmin> configure-session-persistence --user admin --password adminadmin  
--type ha --frequency web-method --scope modified-session --store jdbc/hastore  
--property maxSessions=1000:reapIntervalSeconds=60 server2
```

EXIT STATUS

0 command executed successfully
1 error in executing the command

SEE ALSO

set(1AS), clear-session-store(1AS), create-session-store(1AS)

NAME	create-acl – adds a new access control list file for the named instance
SYNOPSIS	<pre>create-acl --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --aclfile <i>filename</i> <i>acl_ID</i></pre>
DESCRIPTION	Gets the access control lists associated with the named server instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>acl_ID</i> 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	<p>EXAMPLE 1 Using create-acl</p> <pre>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</pre> <p>Where: sampleACL is the name of the ACL created.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	delete-acl(1), list-acl(1)

create-authdb(1)

NAME	create-authdb – adds the new authorized database for the named instance
SYNOPSIS	<pre> create-authdb --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --database <i>database</i> --virtualserver <i>virtualserver_ID</i> [--basedn <i>basedn</i>] [--certmaps <i>certmaps</i>] <i>authdb_ID</i> </pre>
DESCRIPTION	Adds the named authorized database associated with the named server instance.
OPTIONS	<pre> --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. --database user database name in the dbswitch.conf file. --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. --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 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. </pre>
OPERANDS	<i>authdb_id</i> user database name in the virtual server's ACL file.
EXAMPLES	<p>EXAMPLE 1 Using create-authdb</p> <pre> asadmin> create-authdb --user admin --password adminadmin --host fuyako --port 7070 --database default --virtualserver server1 --basedn "o=sun" sampleAuth Created AuthDB with id = sampleAuth </pre> <p>Where sampleAuth is the authdb created.</p>

EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	delete-authdb(1), list-authdbs(1)	

create-auth-realm(1)

NAME	create-auth-realm – adds the new authorized realm for the named instance
SYNOPSIS	<pre> create-auth-realm --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --classname <i>realm_class</i> [--property (name=value) [:name=value] *] <i>auth_realm_name</i> </pre>
DESCRIPTION	Adds the named authorized realm associated with the named server instance.
OPTIONS	<pre> --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 Java class which implements this realm. --property optional attributes name/value paris of provider implementation specific attributes. </pre>
OPERANDS	<i>auth_realm_name</i> name of this realm.
EXAMPLES	<p>EXAMPLE 1 Using create-auth-realm</p> <pre> asadmin> create-auth-realm --user admin --password adminadmin --host bluestar --port 4848 --instance server1 --classname com.ipplanet.ias.security.auth.realm.DB.Database --property defaultuser=admin:Password=admin db Created Auth realm with id = db </pre> <p>Where db is the auth realm created.</p>
EXIT STATUS	<pre> 0 command executed successfully 1 error in executing the command </pre>
SEE ALSO	delete-auth-realm(1), list-auth-realms(1)

NAME	create-custom-resource – registers the custom resource to the named instance																								
SYNOPSIS	<pre>create-custom-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --resourcetype <i>type</i> --factoryclass <i>classname</i> [--enabled=true] [--description <i>text</i>] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>jndi_name</i></pre>																								
DESCRIPTION	Registers the custom resource to the named instance.																								
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--resourcetype</td><td>type of custom resource to be created.</td></tr> <tr> <td>--factoryclass</td><td>class that creates the custom resource.</td></tr> <tr> <td>--enable</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the JDBC connection pool.</td></tr> <tr> <td>--property</td><td>optional attributes name/value pairs for configuring the resource.</td></tr> </table>	--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.	--enable	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.
--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.																								
--enable	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	<i>jndi_name</i> JNDI name of the custom resource to be created.																								
EXAMPLES	<p>EXAMPLE 1 Using create-custom-resources</p> <pre>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</pre> <p>Where sample_custom_resource is the custom resource created.</p>																								
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command																				
0	command executed successfully																								
1	error in executing the command																								

create-custom-resource(1)

SEE ALSO | delete-custom-resource(1), list-custom-resources(1)

NAME	create-domain – creates a domain with the given name
SYNOPSIS	create-domain [--path <i>domain_path</i>] [--sysuser <i>sys_user</i>] [--passwordfile <i>filename</i>] --adminport <i>port_number</i> --adminuser <i>admin</i> --adminpassword <i>password</i> <i>domain_name</i>
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	<div> --path <div>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.</div> </div> <div> --sysuser <div>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.</div> </div> <div> --passwordfile <div>file containing passwords appropriate for the command (e.g., administrative instance).</div> </div> <div> --adminport <div>port of the administrative instance. The port number cannot be currently active.</div> </div> <div> --adminuser <div>user name associated with the administrative instance.</div> </div> <div> --adminpassword <div>password associated with the administrative name.</div> </div>
OPERANDS	<i>domain_name</i> name of the domain. Must be a unique name.
EXAMPLES	<p>EXAMPLE 1 Using create-domain</p> <pre>asadmin> create-domain --path /u/domain1/domain_root --sysuser user1 --adminuser admin --adminpassword adminadmin --adminport 6868 domain1 created domain domain1 successfully</pre> <p>Where: the domain1 domain is created in the domain_root directory.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	delete-domain(1), start-domain(1), stop-domain(1), list-domains(1), multimode(1)

create-file-user(1)

NAME	create-file-user – creates a new file user
SYNOPSIS	<pre>create-file-user --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] [--userpassword <i>user_password</i>] [--groups <i>user_groups:[user_groups]*</i>] <i>user_name</i></pre>
DESCRIPTION	Creates an entry in keyfile by the specified <i>user_name</i> , <i>user_password</i> and groups. Multiple groups can be created by separating them with a colon, ":".
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>user_name</i> name of file user to be created.
EXAMPLES	<p>EXAMPLE 1 Using the create-file-user command to create a file user</p> <pre>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</pre> <p>Where: the <i>sample_user</i> is the file user created.</p> <p>EXAMPLE 2 Using the create-file-user command with the passwordfile option</p> <pre>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</pre> <p>Where: <i>sample_password</i> file contains the following:</p> <pre>AS_ADMIN_PASSWORD=adminadmin AS_ADMIN_USERPASSWORD=sample</pre>

EXAMPLE 2 Using the create-file-user command with the passwordfile option (Continued)

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 delete-file-user(1), list-file-users(1), update-file-user(1),
 list-file-groups(1)

create-http-listener(1)

NAME	create-http-listener – adds a new HTTP listener socket																						
SYNOPSIS	<pre> create-http-listener --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] --address <i>address</i> [--instance <i>instance_name</i>] --listenerport <i>listener_port</i> --defaultvs <i>virtual_server</i> --servername <i>server_name</i> [--family <i>family</i>] [--acceptorthreads <i>acceptor_threads</i>] [--blockingenabled <i>blocking_enabled</i>] [--security <i>enabled</i> <i>security_enabled</i>] [--enabled=<i>enabled</i>] <i>listener_ID</i> </pre>																						
DESCRIPTION	Creates the HTTP listener associated with the named identifier.																						
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--listenerport</td><td>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.</td></tr> <tr> <td>--defaultvs</td><td>ID attribute of the default virtual server for this particular connection group.</td></tr> <tr> <td>--servername</td><td>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.</td></tr> <tr> <td>--family</td><td>socket family type; defaults to <i>inet</i>. Legal values are: <i>inet</i>, <i>inet6</i>, and <i>nca</i>. Use the value <i>inet6</i> for IPv6 listen sockets. When using the value of <i>inet6</i>, IPv4 addresses are prefixed with <i>::ffff:</i> in the log file. Specify <i>nca</i> to make use of the Solaris Network Cache and Accelerator.</td></tr> </table>	--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 <i>inet</i> . Legal values are: <i>inet</i> , <i>inet6</i> , and <i>nca</i> . Use the value <i>inet6</i> for IPv6 listen sockets. When using the value of <i>inet6</i> , IPv4 addresses are prefixed with <i>::ffff:</i> in the log file. Specify <i>nca</i> to make use of the Solaris Network Cache and Accelerator.
--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 <i>inet</i> . Legal values are: <i>inet</i> , <i>inet6</i> , and <i>nca</i> . Use the value <i>inet6</i> for IPv6 listen sockets. When using the value of <i>inet6</i> , IPv4 addresses are prefixed with <i>::ffff:</i> in the log file. Specify <i>nca</i> to make use of the Solaris Network Cache and Accelerator.																						

create-http-listener(1)

	<p>--acceptorthreads number of acceptor threads for the listen socket. The recommended value is the number of processors in the machine.</p> <p>--blockingenabled determines whether the HTTP listener socket and the accepted socket are put into blocking mode. Use of blocking mode may improve benchmark scores.</p> <p>--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 <code>init.conf</code> file globally enables or disables SSL by making certificates available to the server instance. Therefore, security in the <code>init.conf</code> file must be ON or security in the <code>server.xml</code> file does not work.</p> <p>--enabled determines whether the resource is enabled at runtime or not.</p>
OPERANDS	<i>listener_id</i> listener ID of the HTTP listener.
EXAMPLES	<p>EXAMPLE 1 Using create-http-listener</p> <pre>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.ipplanet.com --family inet6 --acceptorthreads 2 --blockingenabled=true --securityenabled=false --enabled=false sampleListener Created HTTP listener with id = sampleListener</pre> <p>Where: <code>sampleListener</code> is the HTTP listener created.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	<code>delete-http-listener(1)</code> , <code>list-http-listeners(1)</code>

create-http-qos(1)

NAME	create-http-qos – creates a new quality of service parameter for the named instance																						
SYNOPSIS	<pre> create-http-qos --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--virtualserver <i>virtual_server_ID</i>] [--bwlimit <i>bwlimit</i>] [--enforcebwlimit=<i>enforce_bw_limit</i>] [--connlimit <i>connection_limit</i>] [--enforceconnlimit=<i>enforce_conn_limit</i>] <i>instance_name</i> </pre>																						
DESCRIPTION	Adds a new quality of service parameter associated with the named server instance.																						
OPTIONS	<table> <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> <tr> <td>--virtualserver</td><td>virtual server ID. It can also be referred to as the variable <code>\$id</code> in an <code>obj.conf</code> file. A virtual server ID cannot begin with a number.</td></tr> <tr> <td>--bwlimit</td><td>maximum bandwidth limit, for the virtual server class or virtual server, in bytes per second. The default is no limit.</td></tr> <tr> <td>--enforcebwlimit</td><td>determines whether the bandwidth limit should be enforced or not.</td></tr> <tr> <td>--connlimit</td><td>maximum number of concurrent connections for the server, virtual server class, or virtual server.</td></tr> <tr> <td>--enforceconnlimit</td><td>determines whether the connection limit should be enforced or not.</td></tr> </table>	--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 <code>\$id</code> in an <code>obj.conf</code> 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.
--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 <code>\$id</code> in an <code>obj.conf</code> 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	<i>instance_name</i> name of the instance.																						
EXAMPLES	<p>EXAMPLE 1 Using create-http-qos</p> <pre> 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 </pre>																						

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

SEE ALSO `delete-http-qos(1)`

create-iiop-listener(1)

NAME	create-iiop-listener – adds the IIOP listener for the named instance
SYNOPSIS	<pre> create-iiop-listener --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --listeneraddress <i>address</i> [--iiopport <i>iiop_port</i>] [--enabled=true] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>listener_ID</i> </pre>
DESCRIPTION	Adds the IIOP listener associated with the named server instance.
OPTIONS	<pre> --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. </pre>
OPERANDS	<i>listener_id</i> unique identifier for the IIOP listener to be created.
EXAMPLES	<p>EXAMPLE 1 Using create-iiop-listener</p> <pre> 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 </pre> <p>Where: <i>sample_iiop_listener</i> is the IIOP listener created.</p>
EXIT STATUS	<pre> 0 command executed successfully 1 error in executing the command </pre>

`create-iiop-listener(1)`

SEE ALSO `delete-iiop-listener(1)`, `list-iiop-listeners(1)`

create-instance(1)

NAME	create-instance – creates an application server instance with the specified instance name																				
SYNOPSIS	<pre>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</pre>																				
DESCRIPTION	<p>You can create a new instance on a local or remote machine. When you run the command locally, you can execute commands even when the admin server is not running. However, when this command is executed in local mode, no validations are made to ensure that the <code>--instanceport</code> is not being used by any existing server instances. To validate this execute the <code>create-instance</code> command in remote mode by providing the <code>--user</code> and <code>--password</code> information. This ensures that the instanceport is not being used by any other server instance.</p> <p>On a 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 <code>--local</code> option. The named instance must not exist within that domain.</p>																				
OPTIONS	<table> <tr> <td><code>--user</code></td><td>administrative user associated for the instance.</td></tr> <tr> <td><code>--password</code></td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td><code>--host</code></td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td><code>--port</code></td><td>administrative port number associated with the administrative host.</td></tr> <tr> <td><code>--sysuser</code></td><td>owner of the domain directory.</td></tr> <tr> <td><code>--domain</code></td><td>name of the domain.</td></tr> <tr> <td><code>--local</code></td><td>determines if the command should delegate the request to administrative instance or run locally.</td></tr> <tr> <td><code>--passwordfile</code></td><td>file containing passwords appropriate for the command (e.g., administrative instance).</td></tr> <tr> <td><code>--secure</code></td><td>if true, uses SSL/TLS to communicate with the administrative instance.</td></tr> <tr> <td><code>--instanceport</code></td><td>port where the instance listens for requests.</td></tr> </table>	<code>--user</code>	administrative user associated for the instance.	<code>--password</code>	administrative password corresponding to the administrative user.	<code>--host</code>	host name of the machine hosting the administrative instance.	<code>--port</code>	administrative port number associated with the administrative host.	<code>--sysuser</code>	owner of the domain directory.	<code>--domain</code>	name of the domain.	<code>--local</code>	determines if the command should delegate the request to administrative instance or run locally.	<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).	<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.	<code>--instanceport</code>	port where the instance listens for requests.
<code>--user</code>	administrative user associated for the instance.																				
<code>--password</code>	administrative password corresponding to the administrative user.																				
<code>--host</code>	host name of the machine hosting the administrative instance.																				
<code>--port</code>	administrative port number associated with the administrative host.																				
<code>--sysuser</code>	owner of the domain directory.																				
<code>--domain</code>	name of the domain.																				
<code>--local</code>	determines if the command should delegate the request to administrative instance or run locally.																				
<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).																				
<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.																				
<code>--instanceport</code>	port where the instance listens for requests.																				
OPERANDS	<code>instance_name</code> name of the instance to be created.																				
EXAMPLES	<p>EXAMPLE 1 Using create-instance in local mode</p> <pre>asadmin> create-instance --domain domain1 --instanceport 8967 --sysuser adminuser server4 Created Instance server4 successfully</pre>																				

EXAMPLE 1 Using create-instance in local mode *(Continued)*

Where: the server4 instance is created under the domain1 domain.

EXAMPLE 2 Using create-instance in remote mode

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

SEE ALSO delete-instance(1), start-instance(1), stop-instance(1),
 restart-instance(1)

create-javamail-resource(1)

NAME	create-javamail-resource – registers the Javamail resource to the named instance																																		
SYNOPSIS	<pre> create-javamail-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --mailhost <i>hostname</i> --mailuser <i>username</i> --fromaddress <i>address</i> [--storeprotocol <i>imap</i>] [--storeprotocolclass <i>com.sun.mail.imapIMAPStore</i>] [--transportprotocol <i>smtp</i>] [--transportprotocolclass <i>com.sun.mail.smtp.SMTPTransport</i>] [--debug=false] [--enabled=true] [--description <i>text</i>] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>jndi_name</i> </pre>																																		
DESCRIPTION	Registers the Javamail resource to the named instance.																																		
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--mailhost</td><td>mail server host.</td></tr> <tr> <td>--mailuser</td><td>mail account user name.</td></tr> <tr> <td>--fromaddress</td><td>email address.</td></tr> <tr> <td>--storeprotocol</td><td>mail server stored protocol.</td></tr> <tr> <td>--storeprotocolclass</td><td>mail server stored protocol class name.</td></tr> <tr> <td>--transportprotocol</td><td>mail server transport protocol.</td></tr> <tr> <td>--transportprotocolclass</td><td>mail server transport protocol class name.</td></tr> <tr> <td>--debug</td><td>if set to true, server startup in debug mode for this resource.</td></tr> <tr> <td>--enable</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the JDBC connection pool.</td></tr> </table>	--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.	--mailhost	mail server host.	--mailuser	mail account user name.	--fromaddress	email address.	--storeprotocol	mail server stored protocol.	--storeprotocolclass	mail server stored protocol class name.	--transportprotocol	mail server transport protocol.	--transportprotocolclass	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.
--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.																																		
--mailhost	mail server host.																																		
--mailuser	mail account user name.																																		
--fromaddress	email address.																																		
--storeprotocol	mail server stored protocol.																																		
--storeprotocolclass	mail server stored protocol class name.																																		
--transportprotocol	mail server transport protocol.																																		
--transportprotocolclass	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.																																		

	create-javamail-resource(1)
	<pre>--property</pre> <p>optional attributes name/value pairs for configuring the resource.</p>
OPERANDS	<pre>jndi_name</pre> <p>JNDI name of the Javamail resource to be created.</p>
EXAMPLES	<p>EXAMPLE 1 Using create-javamail-resource</p> <pre>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</pre> <p>Where: <code>sample_javamail_resource</code> is the javamail resource created. The escape character (\) is used in the <code>fromaddress</code> option to distinguish the dot (.) and @ sign.</p>
EXIT STATUS	<pre>0</pre> <p>command executed successfully</p> <pre>1</pre> <p>error in executing the command</p>
SEE ALSO	<pre>delete-javamail-resource(1), list-javamail-resources(1)</pre>

create-jdbc-connection-pool(1)

NAME	create-jdbc-connection-pool – registers the JDBC connection pool to the named instance																						
SYNOPSIS	<pre> create-jdbc-connection-pool --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --datasourceclassname <i>classname</i> [--restype <i>res_type</i>] [--steadypoolsize <i>8</i>] [--maxpoolsize <i>32</i>] [--maxwait <i>6000</i>] [--poolresize <i>2</i>] [--idletimeout <i>300</i>] [--isolationlevel <i>isolation_level</i>] [--isisolationguaranteed=true] [--isconnectvalidatereq=false] [--validationmethod <i>auto-commit</i>] [--validationtable <i>table_name</i>] [--failconnection=false] [--description <i>text</i>] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>connection_pool_ID</i> </pre>																						
DESCRIPTION	Registers the JDBC connection pool to the named instance.																						
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--datasourceclassname</td><td>name of the vendor supplied JDBC datasource resource manager.</td></tr> <tr> <td>--restype</td><td>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.</td></tr> <tr> <td>--steadypoolsize</td><td>minimum and initial number of connections maintained in the pool.</td></tr> <tr> <td>--maxpoolsize</td><td>maximum number of connections that can be created.</td></tr> </table>	--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.	--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.
--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.																						
--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.																						

create-jdbc-connection-pool(1)

<code>--maxwait</code>	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.
<code>--poolresize</code>	number of connections to be removed when <code>idletimeout</code> timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches <code>steadypoolsize</code> , the connection removal stops.
<code>--idletimeout</code>	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.
<code>--isolationlevel</code>	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.
<code>--isconnectvalidatereq</code>	if set to true connections are validated (checked to see if they are usable) before giving out the application. The default is false.
<code>--validationmethod</code>	name of the validation table used to perform a query to validate a connection.
<code>--validationtable</code>	name of the validation table used to perform a query to validate a connection. This parameter is mandatory if <code>connection-validation-type</code> is set to table. Verification by accessing a user specified table may become necessary for connection validation.
<code>--failconnection</code>	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.
<code>--description</code>	text description of the JDBC connection pool.

create-jdbc-connection-pool(1)

	<code>--property</code>	optional attributes name/value pairs for configuring the connection pool.
OPERANDS	<code>connection_pool_id</code>	name of the JDBC connection pool to be created.
EXAMPLES	<p>EXAMPLE 1 Using create-jdbc-connection-pool</p> <pre>asadmin> create-jdbc-connection-pool --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --datasourceclassname XA --restype javax.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</pre> <p>Where: the XA_connection_pool is created. The escape character "\\" is used in the --property option to distinguish the colons (:) and the backslash (/).</p>	
EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	delete-jdbc-connection-pool(1), list-jdbc-connection-pool(1)	

create-jdbc-resource(1)

NAME	create-jdbc-resource – registers the JDBC resource to the named instance
SYNOPSIS	create-jdbc-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --connectionpoolid <i>ID</i> [--enabled=true] [--description <i>text</i>] <i>jndi_name</i>
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	<i>jndi_name</i> JNDI name of the JDBC resource to be created.
EXAMPLES	EXAMPLE 1 Using the create-jdbc-resource command <pre>asadmin> create-jdbc-resource --usre 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</pre> <p>Where: <i>sample_jdbc_resource</i> is the resource that is created.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	delete-jdbc-resource(1), list-jdbc-resources(1)

create-jmsdest(1)

NAME	create-jmsdest – adds the named destination
SYNOPSIS	<pre>create-jmsdest --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --desttype <i>type</i> [--property (<i>name=value</i>) [:<i>name=value</i>] *] <i>dest_name</i></pre>
DESCRIPTION	Adds the named destination.
OPTIONS	<pre>--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 <i>topic</i>, and <i>queue</i>. --property name/value pairs used for specifying iMQ specific attributes to further customize the destination to be created.</pre>
OPERANDS	<pre><i>dest_name</i> name of the JMS destination. Valid value is any name that can be a Java identifier.</pre>
EXAMPLES	<p>EXAMPLE 1 Using create-jmsdest</p> <pre>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 desttype=topic</pre>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	delete-jmsdest(1), list-jmsdest(1)

NAME	create-jms-resource – registers the JMS resource to the named instance																						
SYNOPSIS	<pre>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</pre>																						
DESCRIPTION	Registers the JMS resource to the named instance.																						
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--resourcetype</td><td>JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.</td></tr> <tr> <td>--enabled</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the JDBC connection pool.</td></tr> <tr> <td>--property</td><td>optional attributes name/value pairs for configuring the JMS resource.</td></tr> </table>	--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	JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.	--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.
--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	JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.																						
--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	<i>jndi_name</i> JNDI name of the JMS resource to be created.																						
EXAMPLES	<p>EXAMPLE 1 Using the create-jms-resource command</p> <pre>asadmin> create-jms-resource --user admin --password adminadmin --host fuyako --port 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</pre> <p>Where: the sample_jms_resource is the resource that is created.</p>																						
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command																		
0	command executed successfully																						
1	error in executing the command																						

create-jms-resource(1)

**INTERFACE
EQUIVALENT**

JMS folder, Destinations page

SEE ALSO

delete-jms-resource(1), list-jms-resources(1)

NAME	create-jndi-resource – registers the JNDI resource to the named instance																										
SYNOPSIS	<pre> create-jndi-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --jndilookupname <i>lookup_name</i> --resourcetype <i>type</i> --factoryclass <i>class_name</i> [--enabled=true] [--description <i>text</i>] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>jndi_name</i> </pre>																										
DESCRIPTION	Registers the JNDI resource to the named instance.																										
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--jndilookupname</td><td>lookup name used by external container.</td></tr> <tr> <td>--resourcetype</td><td>JNDI resource type which can be: topic or queue.</td></tr> <tr> <td>--factoryclass</td><td>class that creates the JNDI resource.</td></tr> <tr> <td>--enabled</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the JDBC connection pool.</td></tr> <tr> <td>--property</td><td>optional attributes name/value pairs for configuring the JNDI resource.</td></tr> </table>	--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.
--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	<i>jndi_name</i> name of the JNDI resource to be created.																										
EXAMPLES	<p>EXAMPLE 1 Using the create-jndi-resource command</p> <pre> 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 </pre> <p>Where: sample_jndi_resource is the JNDI resource created.</p>																										

create-jndi-resource(1)

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

SEE ALSO	delete-jndi-resource(1), list-jndi-resources(1)
-----------------	-------------------------------------------------

NAME	create-jvm-options – creates the JVM options from the Java configuration or profiler elements
SYNOPSIS	<pre>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] *</pre>
DESCRIPTION	Creates the JVM options in the Java configuration or Profiler elements of the <code>server.xml</code> 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	<pre>--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.</pre>
OPERANDS	<pre>jvm_option_name=jvm_option_value</pre> <p>the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the <code>jvm_option_value</code>.</p>
EXAMPLES	<p>EXAMPLE 1 Using create-jvm-options</p> <pre>asadmin> create-jvm-options --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --profiler=false -- "-DDebug=true":"-Xmx256m":"-Dcom.sun.aas.imqBin"="\export\as7se\imq\bin" JVM options created</pre> <p>Where the JVM options are created. The double dash (—) is used between <code>--profiler</code> 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.</p>

create-jvm-options(1)

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

SEE ALSO	delete-jvm-options(1)
-----------------	-----------------------

NAME	create-lifecycle-module – adds a lifecycle module for the named instance																												
SYNOPSIS	<pre> create-lifecycle-module --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --classname <i>class_name</i> [--classpath <i>classpath</i>] [--loadorder <i>load_order</i>] [--failurefatal <i>failure_fatal</i>] [--enabled=true] [--property (<i>name=value</i>)[:<i>name=value</i>]*] <i>module_name</i> </pre>																												
DESCRIPTION	Creates the lifecycle module associated with the named server instance.																												
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--classname</td><td>fully qualified name of the startup class.</td></tr> <tr> <td>--classpath</td><td>indicates where this module is actually located if it is not under applications-root.</td></tr> <tr> <td>--loadorder</td><td>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.</td></tr> <tr> <td>--failurefatal</td><td>if true indicates abort server startup if this module does not load properly.</td></tr> <tr> <td>--enable</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the resource.</td></tr> <tr> <td>--property</td><td>optional attributes name/value pairs for configuring the resource.</td></tr> </table>	--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.	--failurefatal	if true indicates abort server startup if this module does not load properly.	--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.
--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.																												
--failurefatal	if true indicates abort server startup if this module does not load properly.																												
--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	<table> <tr> <td><i>module_name</i></td><td>unique identifier for the deployed server lifecycle event listener module.</td></tr> </table>	<i>module_name</i>	unique identifier for the deployed server lifecycle event listener module.																										
<i>module_name</i>	unique identifier for the deployed server lifecycle event listener module.																												

create-lifecycle-module(1)

EXAMPLES

EXAMPLE 1 using create-lifecycle-module

```
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

0 command executed successfully
1 error in executing the command

SEE ALSO

delete-lifecycle-module(1), list-lifecycle-modules(1)

NAME	create-mime – adds the MIME type for the named instance.
SYNOPSIS	<pre>create-mime --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --mimefile <i>filename</i> <i>mime_ID</i></pre>
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	<pre>--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.</pre>
OPERANDS	<i>mime_id</i> 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	<p>EXAMPLE 1 Using create-mime</p> <pre>asadmin> create-mime --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --mimefile mime.types sampleMIME Created Mime with id = sampleMIME</pre> <p>Where: sampleMIME is the name of the MIME created.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	delete-mime(1), list-mimes(1)

create-persistence-resource(1)

NAME	create-persistence-resource – registers the persistence resource to the named instance																								
SYNOPSIS	<pre>create-persistence-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] [--jdbcjndiname <i>jndi_name</i>] [--factoryclass <i>classname</i>] [--enabled=true] [--description <i>text</i>] [--property (<i>name=value</i>)[:<i>name=value</i>] *] <i>jndi_name</i></pre>																								
DESCRIPTION	Registers the persistence resource associated with the specified JNDI name from the named instance.																								
OPTIONS	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--jdbcjndiname</td><td>JDBC resource with which database connections are obtained. Must be the name of one of the pre-created JDBC resources.</td></tr> <tr> <td>--factoryclass</td><td>class that creates persistence manager instance.</td></tr> <tr> <td>--enable</td><td>determines whether the resource is enabled at runtime or not.</td></tr> <tr> <td>--description</td><td>text description of the resource.</td></tr> <tr> <td>--property</td><td>optional attributes name/value pairs for configuring the resource.</td></tr> </table>	--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.
--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	<i>jndi_name</i> JNDI name of the persistence manager factory resource.																								
EXAMPLES	<p>EXAMPLE 1 Using create-persistence-resource</p> <pre>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</pre> <p>Where: <i>sample_persistence_resource</i> is the persistence manager factory resource created.</p>																								
EXIT STATUS	0 command executed successfully																								

create-persistence-resource(1)

1 error in executing the command

SEE ALSO delete-persistence-resource(1), list-persistence-resources(1)

create-profiler(1)

NAME	create-profiler – creates the profiler element																						
SYNOPSIS	<pre>create-profiler --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --classpath <i>classpath</i> [--nativelibpath <i>native_library_path</i>] [--enabled=true] [--property (<i>name=value</i>) [:<i>name=value</i>] *]<i>profiler_name</i></pre>																						
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	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--classpath</td><td>Java classpath string that specifies the classes needed by the profiler.</td></tr> <tr> <td>--nativelibpath</td><td>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.</td></tr> <tr> <td>--enabled</td><td>profiler is enabled by default.</td></tr> <tr> <td>--property</td><td>name/value pairs of provider specific attributes.</td></tr> </table>	--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.
--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	<i>profiler_name</i> name of the profiler.																						
EXAMPLES	<p>EXAMPLE 1 Using create-profiler</p> <pre>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 <i>sample_profiler</i> Created Profiler with id = <i>sample_profiler</i></pre> <p>Where: <i>sample_profiler</i> is the profiler created.</p>																						

create-profiler(1)

EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	delete-profiler(1), list-profilers(1)	

create-session-store(1)

NAME	create-session-store – creates the session schema in the persistent store										
SYNOPSIS	<pre>create-session-store [--storeurl <i>persistent_store_URL</i> --storeuser <i>username</i> [--storepassword <i>user_password</i>] [--dbssystempassword <i>system_password</i>]] [--optionsfile <i>filename</i>]</pre>										
DESCRIPTION	<p>Enables the user to create the session schema in the persistent store based on:</p> <table> <tr> <td>storeurl</td><td>The JDBC URL that is pointing to the persistent store being used for saving the session data.</td></tr> <tr> <td>storeuser</td><td>The username associated with the persistent store being used for saving session data.</td></tr> <tr> <td>storepassword</td><td>The password corresponding to the user of the persistent store being used for saving session data.</td></tr> <tr> <td>dbssystempassword</td><td>The password for the HADB system user. This is the same password you specify when you create the HADB database using the HADB <code>create</code> command.</td></tr> </table> <p>All the options should be specified either on the command-line itself, or in an options file. If passwords are not specified on the command-line, the system prompts you to provide it, if you are in the interactive mode.</p> <p>This command can only be run locally.</p>	storeurl	The JDBC URL that is pointing to the persistent store being used for saving the session data.	storeuser	The username associated with the persistent store being used for saving session data.	storepassword	The password corresponding to the user of the persistent store being used for saving session data.	dbssystempassword	The password for the HADB system user. This is the same password you specify when you create the HADB database using the HADB <code>create</code> command.		
storeurl	The JDBC URL that is pointing to the persistent store being used for saving the session data.										
storeuser	The username associated with the persistent store being used for saving session data.										
storepassword	The password corresponding to the user of the persistent store being used for saving session data.										
dbssystempassword	The password for the HADB system user. This is the same password you specify when you create the HADB database using the HADB <code>create</code> command.										
OPTIONS	<table> <tr> <td>--storeurl</td><td>JDBC URL of the persistent store.</td></tr> <tr> <td>--storeuser</td><td>username associated for the persistent store.</td></tr> <tr> <td>--storepassword</td><td>password corresponding to the storeuser.</td></tr> <tr> <td>--dbssystempassword</td><td>password corresponding to the HADB system user.</td></tr> <tr> <td>--optionsfile</td><td>a valid filename that has the options in it.</td></tr> </table>	--storeurl	JDBC URL of the persistent store.	--storeuser	username associated for the persistent store.	--storepassword	password corresponding to the storeuser.	--dbssystempassword	password corresponding to the HADB system user.	--optionsfile	a valid filename that has the options in it.
--storeurl	JDBC URL of the persistent store.										
--storeuser	username associated for the persistent store.										
--storepassword	password corresponding to the storeuser.										
--dbssystempassword	password corresponding to the HADB system user.										
--optionsfile	a valid filename that has the options in it.										
EXAMPLES	<p>EXAMPLE 1 Using create-session-store</p> <pre>asadmin> create-session-store --storeurl jdbc:sun:hadb@myhost1:7676,myhost2:6767 --storeuser haadmin --storepassword hapasswd --dbssystempassword super</pre>										
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command						
0	command executed successfully										
1	error in executing the command										
SEE ALSO	configure-session-persistence(1), clear-session-store(1)										

NAME	create-ssl – creates the SSL element in the HTTP listener or IIOP listener																										
SYNOPSIS	<pre> create-ssl --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] --type [http-listener iiop-listener iiop-service] --certname <i>cert_name</i> [--instance <i>instance_name</i>] [--ssl2enabled=false] [--ssl2ciphers <i>ssl_2_ciphers</i>] [--ssl3enabled=true] [--ssl3tlsciphers <i>ssl3_tls_ciphers</i>] [--tlseenabled=true] [--tlscrollbackenabled=true] [--clientauthenabled=false] [<i>listener_id</i>] </pre>																										
DESCRIPTION	Creates the ssl element from the HTTP listener or IIOP listener.																										
OPTIONS	<table> <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>--secure</td><td>indicates communication with the administrative instance in secured mode.</td></tr> <tr> <td>--passwordfile</td><td>file containing passwords appropriate for the command (e.g., administrative instance).</td></tr> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--type</td><td>type of service or listener that the SSL is created for. The type can be: http-listener, iiop-listener, and iiop-service.</td></tr> <tr> <td>--certname</td><td>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.</td></tr> <tr> <td>--ssl2enabled</td><td>determines whether SSL2 is enabled.</td></tr> <tr> <td>--ssl2ciphers</td><td>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.</td></tr> <tr> <td>--ssl3enabled</td><td>determines whether SSL3 is enabled.</td></tr> <tr> <td>--ssl3ciphers</td><td>a comma separated list of the SSL3 ciphers used. Use the prefix + to enable or - to disable. Allowed values</td></tr> </table>	--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
--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																										

create-ssl(1)

	are: rsa_rc4_128_md5, rsa_3des_sha, rsa_des_sha, rsa_rc4_40_md5, rsa_rc2_40_md5, rsa_null_md5. Allowed TLS 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	<i>listener_ID</i> the ID of the listener or service that the SSL is created for.
EXAMPLES	<p>EXAMPLE 1 Using create-ssl</p> <pre>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 --ssl3tlsciphers +rsa_rc4_128_md,+rsa_3des_sha,+rsa_des_sha,+rsa_rc4_40_md5 --tlsenabled=false --tlsrollbackenabled=false --clientauthenabled=false http-listener-1</pre> <p>Created SSL in HTTP Listener</p> <p>Where: SSL is created for http-listener-1.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	delete-ssl(1)

NAME	create-virtual-server – adds the named virtual server																								
SYNOPSIS	<pre> create-virtual-server --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] --hosts <i>hosts</i> --mime <i>mime_types_file</i> [--httplisteners <i>http_listeners</i>] [--defaultwebmodule <i>default_web_module</i>] [--configfile <i>config_file</i>] [--defaultobj <i>default_object</i>] [--state <i>on</i>] [--acls <i>acls</i>] [--acceptlang=<i>false</i>] [--logfile <i>log_file</i>] [--property (<i>name=value</i>) [:<i>name=value</i>]*] <i>virtual_server_ID</i> </pre>																								
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	<table> <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> <tr> <td>--instance</td><td>name of the instance.</td></tr> <tr> <td>--hosts</td><td>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.</td></tr> <tr> <td>--mime</td><td>the ID of the mime element used by the virtual server.</td></tr> <tr> <td>--httplisteners</td><td>optional; a comma separated (,) list of HTTP listener IDs. Required only for a virtual server that is not the default virtual server.</td></tr> <tr> <td>--defaultwebmodule</td><td>standalone web module associated with this virtual server by default.</td></tr> <tr> <td>--configfile</td><td>typically all virtual server initialization is from <code>\$INSTANCE_ROOT/config/obj.conf</code>. This can be changed using this attribute.</td></tr> </table>	--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 <code>\$INSTANCE_ROOT/config/obj.conf</code> . This can be changed using this attribute.
--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 <code>\$INSTANCE_ROOT/config/obj.conf</code> . This can be changed using this attribute.																								

create-virtual-server(1)

	--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	<i>virtual_server_id</i>	identifies the unique ID for the virtual server to be created. This virtual server ID cannot begin with a number.
EXAMPLES	<p>EXAMPLE 1 Using create-virtual-server</p> <pre>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</pre> <p>Where sample_vs1 is the virtual server created.</p>	
EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	delete-virtual-server(1), list-virtual-servers(1)	

NAME	delete-acl – removes the access control list file for the named instance
SYNOPSIS	<pre>delete-acl --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>acl_ID</i></pre>
DESCRIPTION	Gets the access control lists associated with the named server instance..
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>acl_ID</i> 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	<p>EXAMPLE 1 Using delete-acl</p> <pre>asadmin> delete-acl --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sampleACL Deleted ACL with id = sampleACL</pre> <p>Where: sampleACL is the ACL that is deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-acl(1), list-acl(1)

delete-authdb(1)

NAME	delete-authdb – removes the authorized database for the named instance
SYNOPSIS	<pre>delete-authdb --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] --virtualserver virtualserver_ID authdb_ID</pre>
DESCRIPTION	Removes the authorized database associated with the named server instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>authdb_id</i> user database name in the virtual server's ACL file.
EXAMPLES	<p>EXAMPLE 1 Using delete-authdb</p> <pre>asadmin> delete-authdb --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --virtualserver server1 sampleAuth Deleted AuthDB with id = sampleAuth</pre> <p>Where: sampleAuth is the authdb deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-authdb(1), list-authdbs(1)

NAME	delete-auth-realm – removes the named authorized realm
SYNOPSIS	delete-auth-realm --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>auth_realm_name</i>
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	<i>auth_realm_name</i> name of this realm.
EXAMPLES	EXAMPLE 1 Using delete-auth-realm <pre>asadmin> delete-auth-realm --user admin --passwordfile passwords.txt --host localhost --port 4848 --instance server1 db Deleted Auth realm with id = db</pre> <p>Where db is the auth realm deleted.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-auth-realm(1), list-auth-realms(1)

delete-custom-resource(1)

NAME	delete-custom-resource – removes the custom resource from the named instance
SYNOPSIS	delete-custom-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>jndi_name</i>
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	<i>jndi_name</i> JNDI name of the custom resource to be deleted.
EXAMPLES	EXAMPLE 1 Using delete-custom-resource <pre>asadmin> delete-custom-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_custom_resource</pre> 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
SEE ALSO	create-custom-resource(1), list-custom-resources(1)

NAME	delete-domain – deletes the given domain
SYNOPSIS	delete-domain <i>domain_name</i>
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	<i>domain_name</i> name of the domain; must be a unique name.
EXAMPLES	EXAMPLE 1 Using delete-domain <pre>asadmin> delete-domain domain1 deleted domain domain1 successfully</pre> <p>Where: the domain1 domain is deleted.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-domain(1), start-domain(1), stop-domain(1), list-domains(1), multimode(1)

delete-file-user(1)

NAME	delete-file-user – removes the named file user
SYNOPSIS	delete-file-user --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>user_name</i>
DESCRIPTION	Deletes an entry in keyfile by the <i>user_name</i> .
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	<i>user_name</i> name of file user to be deleted.
EXAMPLES	EXAMPLE 1 Using the delete-file-user command asadmin> delete-file-user --user admin --password adminadmin --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	create-file-user(1), list-file-users(1), update-file-user(1), list-file-groups(1)

delete-http-listener(1)

NAME	delete-http-listener – removes the HTTP listener for the named instance
SYNOPSIS	delete-http-listener --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>httplistener_ID</i>
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	<i>listener_id</i> listener ID of the HTTP listener.
EXAMPLES	EXAMPLE 1 Using delete-http-listener <pre>asadmin> delete-http-listener --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sampleListener Deleted HTTP listener with id = sampleListener</pre> <p>Where: sampleListener is the HTTP listener deleted.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-http-listener(1), list-http-listeners(1)

delete-http-qos(1)

NAME	delete-http-qos – removes the quality of service parameter for the named instance
SYNOPSIS	delete-http-qos --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--virtualserver <i>virtual_server_ID</i>] <i>instance_name</i>
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 <i>\$id</i> in an <i>obj.conf</i> file. A virtual server ID cannot begin with a number.
OPERANDS	<i>instance_name</i> 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
SEE ALSO	create-http-qos(1)

delete-iiop-cluster-endpoint(1)

NAME	delete-iiop-cluster-endpoint – deletes an IIOP endpoint from the IIOP cluster																		
SYNOPSIS	<pre>delete-iiop-cluster-endpoint [--host admin-host] [--port admin_port] [--user admin-username] [--password admin-password] [--passwordfile filename] [--secure -s] [--iiopserverinstance iiop-server-instance] [--instance instance_name] iiop-endpoint-id</pre>																		
DESCRIPTION	<p>Deletes an IIOP endpoint from the IIOP cluster. For the changes to take effect, the instance must be reconfigured and then restarted after executing this command. Specify the server instance that must be deleted from the IIOP configuration, the IIOP endpoint of the IIOP cluster configuration, and the instance to where the operation is directed.</p> <p>If both the <code>--iiopserverinstance</code> and the <code>--iiopendpointid</code> are specified, the endpoint is removed if at least one endpoint remains after the deletion. If there are no remaining IIOP endpoints, the endpoints and the server is removed.</p> <p>If the <code>--iiopserverinstance</code> alone is specified, then the instance is removed from the IIOP cluster configuration along with all its IIOP endpoint entries.</p> <p>If neither the <code>--iiopserverinstance</code> nor the <code>--iiopendpointid</code> are specified, the following message is displayed:</p> <pre>All the iiopendpoints will be deleted. Are you sure (Y/N)?</pre> <p>If the response is affirmative, the entire cluster configuration is deleted.</p> <p>The <code>delete-iiop-cluster-endpoint</code> command is available only in the <i>Enterprise Edition</i> of the Sun Java System Application Server.</p>																		
OPTIONS	<table> <tr> <td><code>-H --host</code></td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td><code>-p --port</code></td><td>administrative port number associated with the administrative host.</td></tr> <tr> <td><code>-u --user</code></td><td>administrative user associated for the instance.</td></tr> <tr> <td><code>-w --password</code></td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td><code>--passwordfile</code></td><td>file containing passwords appropriate for the command (for example, administrative instance).</td></tr> <tr> <td><code>-s --secure</code></td><td>indicates communication with the administrative instance in secured mode.</td></tr> <tr> <td><code>--iiopserverinstance</code></td><td>name of the server instance to be deleted from the IIOP cluster.</td></tr> <tr> <td><code>--iiopendpointid</code></td><td>identification of IIOP endpoint.</td></tr> <tr> <td><code>--instance</code></td><td>name of the server instance at which this operation is targeted.</td></tr> </table>	<code>-H --host</code>	host name of the machine hosting the administrative instance.	<code>-p --port</code>	administrative port number associated with the administrative host.	<code>-u --user</code>	administrative user associated for the instance.	<code>-w --password</code>	administrative password corresponding to the administrative user.	<code>--passwordfile</code>	file containing passwords appropriate for the command (for example, administrative instance).	<code>-s --secure</code>	indicates communication with the administrative instance in secured mode.	<code>--iiopserverinstance</code>	name of the server instance to be deleted from the IIOP cluster.	<code>--iiopendpointid</code>	identification of IIOP endpoint.	<code>--instance</code>	name of the server instance at which this operation is targeted.
<code>-H --host</code>	host name of the machine hosting the administrative instance.																		
<code>-p --port</code>	administrative port number associated with the administrative host.																		
<code>-u --user</code>	administrative user associated for the instance.																		
<code>-w --password</code>	administrative password corresponding to the administrative user.																		
<code>--passwordfile</code>	file containing passwords appropriate for the command (for example, administrative instance).																		
<code>-s --secure</code>	indicates communication with the administrative instance in secured mode.																		
<code>--iiopserverinstance</code>	name of the server instance to be deleted from the IIOP cluster.																		
<code>--iiopendpointid</code>	identification of IIOP endpoint.																		
<code>--instance</code>	name of the server instance at which this operation is targeted.																		

delete-iiop-cluster-endpoint(1)

OPERANDS None

EXAMPLES

EXAMPLE 1 Removing an IIOP server instance along with all its endpoints from the IIOP cluster of server1

```
asadmin> delete-iiop-cluster-endpoint --user admin --password myPasswd
--iiopserverinstance server2 --instance server1
Deleted instance server2 from the IIOP cluster of server1.
```

EXAMPLE 2 Removing an IIOP endpoint, endpoint1 of an IIOP instance server2 from the IIOP cluster of server1

```
asadmin> delete-iiop-cluster-endpoint --user admin --password myPasswd
--iiopserverinstance server2 --iiopendpointid endpoint1 --instance server1
Deleted IIOP end point endpoint1 of instance server2 from the IIOP cluster of server1.
```

EXAMPLE 3 Removing all IIOP endpoints from the IIOP cluster of server1

```
asadmin> delete-iiop-cluster-endpoint --user admin --password myPasswd
--instance server1
IIOP endpoint(s) deleted successfully
```

SEE ALSO add-iiop-cluster-endpoint(1), list-iiop-cluster-config(1)

NAME	delete-iiop-listener – removes the IIOP listener for the named instance
SYNOPSIS	<pre>delete-iiop-listener --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] listener_ID</pre>
DESCRIPTION	Removes the IIOP listener associated with the named server instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>listener_id</i> unique identifier for the IIOP listener to be deleted.
EXAMPLES	<p>EXAMPLE 1 Using delete-iiop-listener</p> <pre>asadmin> delete-iiop-listener --user admin --password adminadmin --host fuyako --port 7070 sample_iiop_listener Deleted IIOP listener with id = sample_iiop_listener</pre> <p>Where: <i>sample_iiop_listener</i> is the IIOP listener deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-iiop-listener(1), list-iiop-listeners(1)

delete-instance(1)

NAME	delete-instance – deletes the instance that is not running.																
SYNOPSIS	<pre>delete-instance [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--domain domain_name] [--local=false] [--passwordfile filename] [--secure -s] instance_name</pre>																
DESCRIPTION	<p>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.</p> <p>The delete-instance command does not remove the /var/imq/instances/<instance_name> directory.</p>																
OPTIONS	<table> <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>--domain</td><td>name of the domain.</td></tr> <tr> <td>--local</td><td>determines if the command should delegate the request to administrative instance or run locally.</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, SSL/TLS to communicate with the administrative instance.</td></tr> </table>	--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.
--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	<i>instance_name</i> name of the instance to be deleted.																
EXAMPLES	<p>EXAMPLE 1 Using delete-instance in local mode</p> <pre>asadmin> delete-instance --domain domain1 server1 Deleted Instance server1 successfully</pre> <p>Where: the server1 instance for the domain1 domain is deleted on the local machine.</p> <p>EXAMPLE 2 Using delete-instance in remote mode</p> <pre>asadmin> delete-instance --user admin --passwordfile passwords.txt --host localhost --port 4848 server1</pre>																

EXAMPLE 2 Using delete-instance in remote mode *(Continued)*

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 0 command executed successfully
 1 error in executing the command

SEE ALSO create-instance(1), start-instance(1), stop-instance(1),
 restart-instance(1)

delete-javamail-resource(1)

NAME	delete-javamail-resource – removes the Javamail resource from the named instance
SYNOPSIS	delete-javamail-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>jndi_name</i>
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	<i>jndi_name</i> JNDI name of the Javamail resource to be deleted.
EXAMPLES	EXAMPLE 1 Using delete-javamail-resource <pre>asadmin> delete-javamail-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_javamail_resource</pre> 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
SEE ALSO	create-javamail-resource(1), list-javamail-resources(1)

delete-jdbc-connection-pool(1)

NAME	delete-jdbc-connection-pool – removes the JDBC connection pool from the named instance
SYNOPSIS	<pre>delete-jdbc-connection-pool --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] connection_pool_ID</pre>
DESCRIPTION	Removes the JDBC resource from the named instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>connection_pool_id</i> name of the JDBC connection pool to be created.
EXAMPLES	<p>EXAMPLE 1 Using the delete-jdbc-connection-pool command</p> <pre>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</pre> <p>Where: the XA_connection_pool resource is deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-jdbc-connection-pool(1), list-jdbc-connection-pools(1)

delete-jdbc-resource(1)

NAME	delete-jdbc-resource – removes the JDBC resource from the named instance
SYNOPSIS	delete-jdbc-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>jndi_name</i>
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	<i>jndi_name</i> name of the JDBC resource to be deleted.
EXAMPLES	EXAMPLE 1 Using the delete-jdbc-resource command <pre>asadmin> delete-jdbc-resource --user admin --password adminadmin --host fuyako --port 7070 instance server1 sample_jdbc_resource</pre> 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
SEE ALSO	create-jdbc-resource(1), list-jdbc-resources(1)

NAME	delete-jmsdest – destroys the named destination
SYNOPSIS	delete-jmsdest --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--passwordfile <i>filename</i>] [--secure -s] --instance <i>instance_name</i> --desttype <i>type</i> <i>dest_name</i>
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 <i>topic</i> , and <i>queue</i> .
OPERANDS	<i>dest_name</i> name of the JMS destination. Valid value is any name that can be a Java identifier.
EXAMPLES	EXAMPLE 1 Using delete-jmsdest <pre>asadmin> delete-jmsdest --user admin --password adminadmin --host localhost --port 4848 --instance server1 --desttype topic topic_dest</pre> Deleted the JMS Destination with desttype=topic
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-jmsdest(1), list-jmsdest(1)

delete-jms-resource(1)

NAME	delete-jms-resource – removes the JMS resource from the named instance
SYNOPSIS	<pre>delete-jms-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] jndi_name</pre>
DESCRIPTION	Removes the JMS resource from the named instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>jndi_name</i> JNDI name of the JMS resource to be deleted.
EXAMPLES	<p>EXAMPLE 1 Using the delete-jms-resource command</p> <pre>asadmin> delete-jms-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_jms_resource Deleted the JMS resource with jndiname = sample_jms_resource</pre> <p>Where: sample_jms_resource is the resource that is deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-jms-resource(1), list-jms-resources(1)

NAME	delete-jndi-resource – removes the JNDI resource from the named instance
SYNOPSIS	delete-jndi-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] jndi_name
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	<i>jndi_name</i> name of the JNDI resource to be deleted.
EXAMPLES	<p>EXAMPLE 1 Using the delete-jndi-resource command</p> <pre>asadmin> delete-jndi-resource --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sample_jndi_resource</pre> <p>Created the JNDI resource with jndiname = sample_jndi_resource</p> <p>Where: sample_jndi_resource is the JNDI resource to be deleted.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-jndi-resource(1), list-jndi-resources(1)

delete-jvm-options(1)

NAME	delete-jvm-options – deletes the JVM options from the Java configuration or profiler elements																	
SYNOPSIS	delete-jvm-options --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--secure -s] [--instance <i>instance_name</i>] [--profiler=false] (<i>jvm_option_name=jvm_option_value</i>) [: <i>jvm_option_name=jvm_option_name</i>]*																	
DESCRIPTION	Deletes the JVM options in the Java configuration or Profiler elements of the <code>server.xml</code> 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	<table><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><tr><td>--instance</td><td>name of the instance.</td></tr><tr><td>--profiler</td><td>indicates if the JVM options is for the profiler. Profiler must exist for this option to be true.</td></tr></table>		--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.
--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	<i>jvm_option_name=jvm_option_value</i>	the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the <i>jvm_option_value</i> .																
EXAMPLES	<p>EXAMPLE 1 Using delete-jvm-options</p> <pre>asadmin> delete-jvm-options --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --profiler=true -- "-DDebug=true":"-Xmx256m":"-Dcom.sun.aas.imqBin"="\export\as7se\imq\bin" JVM options deleted</pre> <p>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.</p>																	

EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	create-jvm-options(1)	

delete-lifecycle-module(1)

NAME	delete-lifecycle-module – removes the lifecycle module for the named instance
SYNOPSIS	<pre>delete-lifecycle-module --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>module_name</i></pre>
DESCRIPTION	Removes the lifecycle module associated with the named server instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>module_name</i> unique identifier for the deployed server lifecycle event listener module.
EXAMPLES	<p>EXAMPLE 1 Using delete-lifecycle-module</p> <pre>asadmin> delete-lifecycle-module --user admin --password adminadmin --host fuyako --port 7070 customSetup Deleted the Lifecycle module with module name = customSetup</pre> <p>Where: customSetup is the lifecycle module deleted.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-lifecycle-module(1), list-lifecycle-modules(1)

NAME	delete-mime – removes the MIME type for the named instance
SYNOPSIS	delete-mime --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile filename] [--secure -s] [--instance <i>instance_name</i> <i>mime_ID</i>
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 <code>ObjectType</code> section of the <code>obj.conf</code> 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	<i>mime_id</i> 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 <pre>asadmin> delete-mime --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sampleMIME Deleted Mime with id = sampleMIME</pre> <p>Where: <code>sampleMIME</code> is the name of the MIME deleted.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	<code>create-mime(1)</code> , <code>list-mimes(1)</code>

delete-persistence-resource(1)

NAME	delete-persistence-resource – removes the persistence resource from the named instance
SYNOPSIS	delete-persistence-resource --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>jndi_name</i>
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	<i>jndi_name</i> JNDI name of the persistence manager factory resource.
EXAMPLES	EXAMPLE 1 Using delete-persistence-resource <pre>asadmin> delete-persistence-resource --user admin --password adminadmin --host fuyako --port 7070 --instanceserver1 sample_persistence_resource</pre> 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
SEE ALSO	create-persistence-resource(1), list-persistence-resources(1)

NAME	delete-profiler – deletes the profiler element
SYNOPSIS	delete-profiler --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using delete-profiler <pre>asadmin> delete-profiler --user admin --passwordfile passwords.txt --host localhost --port 4848 server1 Deleted Profiler</pre> <p>Where: profiler is deleted from instance server1.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-profiler(1), list-profilers(1)

delete-ssl(1)

NAME	delete-ssl – deletes the ssl element from the HTTP listener or IIOP listener
SYNOPSIS	<pre>delete-ssl --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] --type [<i>http-listener</i> <i>iiop-listener</i> <i>iiop-service</i>] [--instance <i>instance_name</i>] [<i>listener_id</i>]</pre>
DESCRIPTION	Deletes the ssl element from the HTTP listener or IIOP listener.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>listener_ID</i> the ID of the listener or service that the SSL is created for.
EXAMPLES	<p>EXAMPLE 1 Using delete-ssl</p> <pre>asadmin> delete-ssl --user admin --password adminadmin --host fuyako --port 7070 --type http-listener --instance server1 http-listener-1 Deleted SSL in HTTP Listener</pre> <p>Where: SSL is deleted for http-listener-1.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-ssl(1)

NAME	delete-virtual-server – deletes the virtual server with the named virtual server ID
SYNOPSIS	delete-virtual-server --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile filename] [--secure -s] [--instance <i>instance_name</i>] 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	<i>virtual_server_id</i> 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 <pre>asadmin> delete-virtual-server --user admin --password adminadmin --host localhost --port 4848 --instance server1 sample_vs1</pre> 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
SEE ALSO	create-virtual-server(1), list-virtual-servers(1)

deploy(1)

NAME	deploy – deploys the specified component																										
SYNOPSIS	<pre> deploy --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--secure -s] [--passwordfile <i>filename</i>] [--virtualservers <i>virtual_servers</i>] [--type <i>application ejb web connector</i>] [--contextroot <i>context_root</i>] [--force=true] [--precompilejsp=false] [--verify=false] [--name <i>component_name</i>] [--upload=true] [--availabilityenabled] [--retrieve <i>local_dirpath</i>] [--instance <i>instance_name</i>] <i>filepath</i> </pre>																										
DESCRIPTION	<p>Use the deploy command to deploy an EJB, web, client, connector or application. If there is no component with the name you specify, the system returns that the component does not exist. If the component is already deployed or already exists, it is forcefully re-deployed if the --force option is set to true.</p>																										
OPTIONS	<table> <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 (for example, administrative instance).</td></tr> <tr> <td>--secure</td><td>if true, uses SSL/TLS to communicate with the administrative instance.</td></tr> <tr> <td>--virtualservers</td><td>comma separated list of virtual server IDs.</td></tr> <tr> <td>--type</td><td>identifies the type of component to be deployed; defaults to the type of the extension of file.</td></tr> <tr> <td>--contextroot</td><td>valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.</td></tr> <tr> <td>--force</td><td>makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.</td></tr> <tr> <td>--precompilejsp</td><td>by default is set to false which does not allow the JSP to pre-compile during deployment. Instead JSPs are compiled during runtime.</td></tr> <tr> <td>--verify</td><td>the syntax and semantics of the deployment descriptor is verified if set to true.</td></tr> <tr> <td>--name</td><td>name of the deployable component.</td></tr> </table>	--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 (for example, 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.
--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 (for example, 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.																										

deploy(1)

	--upload	when set to true uploads the deployable file to the administration server. If the filepath of the deployable file is mounted to the server machine, or if the administration server is running locally, set the upload option to false.
	--availabilityenabled	valid only for deployment of application, ejb-module and web-module. Valid values are null, true and false. If this option is not specified, it will be considered null and not false. When set to true, availability is enabled for that application or module. For applications, both HTTPSessions and stateful session beans (SFSB) are enabled for availability. The parent web and ejb container values take precedence. The ejb-module or web-module is enabled for availability. The parent ejb or web container value takes precedence. This option is available only in the <i>Enterprise Edition of Sun Java System Application Server</i> .
	--retrieve	retrieves the client stub JAR file from the server machine to the local directory.
	--instance	name of the instance.
OPERANDS	<i>filepath</i>	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.
EXAMPLES	<p>EXAMPLE 1 Using deploy for WAR module</p> <pre>asadmin> deploy --user admin --passwordfile passwords.txt --host localhost --port 4848 --virtualservers server1 --type web --contextroot simple --force=false --precompilejsp=false --verify=false --name simple --upload=true --instance server1 /export/samples/simple.war Deployed the WAR module:simple</pre> <p>Where: the simple WAR module is deployed to the absolute filepath specified.</p> <p>EXAMPLE 2 Using deploy for an application</p> <pre>asadmin> deploy --user admin --password adminadmin --host localhost --port 4848 --virtualservers server1 --type application --force=false --verify=false --name fortune --upload=true --instance server1 /export/samples/fortune.ear Deployed the application:fortune</pre> <p>Where: the fortune application is deployed to the absolute filepath specified.</p>	
EXIT STATUS	0	command executed successfully
	1	error in executing the command

deploy(1)

SEE ALSO deploydir(1), undeploy(1), enable(1), disable(1), list-components(1)

NAME	deploydir – deploys the J2EE component that is in the directory located on the server machine																				
SYNOPSIS	<pre> deploydir --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--virtualservers <i>virtual_servers</i>] [--type <i>application ejb web connector</i>] [--contextroot <i>context_root</i>] [--force=true] [--precompilejsp=false] [--verify=false] [--name <i>component_name</i>] [--availabilityenabled] [--instance <i>instancename</i>] <i>dirpath</i> </pre>																				
DESCRIPTION	<p>Use the <code>deploydir</code> command to deploy the J2EE component that is in the directory located on the server machine. If you use the <code>asadmin deploydir</code> command to deploy a directory instead of an EAR file, your directory structure holding the individual modules must be named with <code>_jar</code>, <code>_war</code> and <code>_rar</code> suffixes.</p> <p>The <code>--force</code> option makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists. Set <code>--force</code> to <code>false</code> for a first deployment. If the application with that name is running, and force is set to <code>false</code>, the command fails.</p>																				
OPTIONS	<table> <tr> <td><code>--user</code></td><td>administrative user associated for the instance.</td></tr> <tr> <td><code>--password</code></td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td><code>--host</code></td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td><code>--port</code></td><td>administrative port number associated with the administrative host.</td></tr> <tr> <td><code>--passwordfile</code></td><td>file containing passwords appropriate for the command (e.g., administrative instance).</td></tr> <tr> <td><code>--secure</code></td><td>if true, uses SSL/TLS to communicate with the administrative instance.</td></tr> <tr> <td><code>--virtualservers</code></td><td>comma separated list of virtual server IDs.</td></tr> <tr> <td><code>--type</code></td><td>type of component to be deployed.</td></tr> <tr> <td><code>--contextroot</code></td><td>valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.</td></tr> <tr> <td><code>--force</code></td><td>makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.</td></tr> </table>	<code>--user</code>	administrative user associated for the instance.	<code>--password</code>	administrative password corresponding to the administrative user.	<code>--host</code>	host name of the machine hosting the administrative instance.	<code>--port</code>	administrative port number associated with the administrative host.	<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).	<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.	<code>--virtualservers</code>	comma separated list of virtual server IDs.	<code>--type</code>	type of component to be deployed.	<code>--contextroot</code>	valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.	<code>--force</code>	makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.
<code>--user</code>	administrative user associated for the instance.																				
<code>--password</code>	administrative password corresponding to the administrative user.																				
<code>--host</code>	host name of the machine hosting the administrative instance.																				
<code>--port</code>	administrative port number associated with the administrative host.																				
<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).																				
<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.																				
<code>--virtualservers</code>	comma separated list of virtual server IDs.																				
<code>--type</code>	type of component to be deployed.																				
<code>--contextroot</code>	valid only if the archive is a web module. It is ignored for other archive types; defaults to filename without extension.																				
<code>--force</code>	makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists.																				

deploydir(1)

	--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.
	--availabilityenabled	Valid only for deployment of application, EJB module, and Web module. Valid values are null, true, and false. If this option is not specified, it will be considered null and not false. When set to true, availability is enabled for that application or module. For applications, both HTTPSessions and stateful session beans (SFSB) are enabled for availability. The parent web and EJB container values take precedence. The EJB module or Web module is enabled for availability. The parent EJB or Web container value take precedence. This option is available only in the Enterprise Edition of Sun Java System Application Server.
	--instance	name of the instance.
OPERANDS	<i>dirpath</i>	path to the directory containing the exploded format of the deployable archive.
EXAMPLES	<p>EXAMPLE 1 Using deploydir</p> <pre>asadmin> deploydir --user admin --passwordfile passwords.txt --host localhost --port 4848 --force=true --verify=false --name fortune --type application --instance server1 /export/samples/fortune Deployed the application:fortune</pre> <p>Where: the fortune application is deployed to the directory specified.</p>	
EXIT STATUS	0	command executed successfully
	1	error in executing the command
SEE ALSO	deploy(1), undeploy(1), enable(1), disable(1), list-components(1)	

NAME	disable – stops the specified component
SYNOPSIS	<pre>disable --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--type application ejb web connector] [--instance <i>instance_name</i>] <i>component_name</i></pre>
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	<pre>--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.</pre>
OPERANDS	<i>component_name</i> name of the component to be disabled.
EXAMPLES	<p>EXAMPLE 1 Using disable</p> <pre>asadmin> disable --user admin --passwordfile passwords.txt --host localhost --port 4848 --type web --instance server 1 simple Disabled the WAR module:simple</pre>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	deploy(1), deploydir(1), undeploy(1), enable(1)

display-license(1)

NAME	display-license – displays the license information
SYNOPSIS	display-license [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--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	<p>EXAMPLE 1 Using display-license in local mode</p> <pre>asadmin> display-license ***** Eval Sun Java System Application Server 7.1 Evaluation License Expiration date Tues 11 Sept 11:58:47 PDT 2002 Number of instances per admin server Unlimited Allow remote administration YES *****</pre> <p>EXAMPLE 2 Using display-license in remote mode</p> <pre>asadmin> display-license --user admin --password adminadmin --host fuyako --port 7070 ***** Eval Sun Java System Application Server 7.1 Evaluation License Expiration date Tues 11 Aug 11:58:47 PDT 2004 Number of instances per admin server Unlimited Allow remote administration YES *****</pre>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	install-license(1)

NAME	enable – runs the specified component
SYNOPSIS	enable --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--secure -s] [--type application ejb web connector] [--instance <i>instance_name</i>] <i>component_name</i>
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	<i>component_name</i> name of the component to be enabled.
EXAMPLES	EXAMPLE 1 Using enable <pre>asadmin> enable --user admin --passwordfile passwords.txt --host localhost --port 4848 --type web --instance server1 simple Enabled the WAR module: simple</pre> <p>Where: the simple WAR module is enabled.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	deploy(1), deploydir(1), undeploy(1), disable(1)

export(1)

NAME	export – marks a variable name for automatic export to the environment of subsequent commands in multimode
SYNOPSIS	export [<i>name=value</i> [<i>name=value</i>] *]
DESCRIPTION	Use the <code>export</code> 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 <code>unset</code> them or exit multimode. If only the variable name is specified, subsequent commands receive a value set in a previous assignment. If the <code>export</code> 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 <code>asadmin</code> utility are imported automatically and set as exported variables within <code>asadmin</code> . Unexported environment variables cannot be read by the <code>asadmin</code> utility.
OPERANDS	<i>name=value</i> variable name and value for automatic export to the environment to be used by subsequent commands.
EXAMPLES	<p>EXAMPLE 1 Using <code>export</code> to set a single environment variable</p> <pre>asadmin> export AS_ADMIN_HOST=bluestar</pre> <p>EXAMPLE 2 Using <code>export</code> to set multiple environment variables</p> <pre>asadmin> export AS_ADMIN_HOST=bluestar AS_ADMIN_PORT=8000 AS_ADMIN_USER=admin AS_ADMIN_PASSWORD=password AS_ADMIN_PREFIX=server1.jms-service</pre> <p>EXAMPLE 3 Using <code>export</code> to list the environment variables</p> <pre>asadmin> export AS_ADMIN_HOST=bluestar AS_ADMIN_PORT=8000 AS_ADMIN_USER=admin AS_ADMIN_PASSWORD=***** AS_ADMIN_PREFIX=server1.jms-service</pre> <p>Where: the <code>export</code> command lists the environment variables that are set. In this case, the environment variables have been set to: the host is <i>bluestar</i>, the port is <i>8000</i>, the administrator user is <i>admin</i> with an associated password, and the prefix is <i>server1.jms-service</i>.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	<code>unset(1)</code> , <code>multimode(1)</code>

NAME	flexanlg – analyzes access log files																						
SYNOPSIS	flexanlg -i <i>filename</i> [-P] [-n <i>servername</i>] [-x] [-r] [-p <i>order</i>] * [-m <i>metafile</i>] * [-o <i>filename</i>] [-c <i>options</i>] [-t <i>options</i>] [-l <i>options</i>]																						
DESCRIPTION	<p>Use the <code>flexanlg</code> 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.</p> <p>These statistics are generated from the server's access log which, by default, is named <code>access</code> and is found in the <code>logs</code> directory of the server instance.</p> <p>Before running the log analyzer, you should archive the server logs.</p>																						
OPTIONS	<p>Options marked with * can be repeated.</p> <table> <tr> <td>-i <i>filename</i></td><td>input log file(s)</td></tr> <tr> <td>-P</td><td>proxy log format</td></tr> <tr> <td>-n <i>servername</i></td><td>the name of the server</td></tr> <tr> <td>-x</td><td>output in HTML</td></tr> <tr> <td>-r</td><td>resolve IP addresses to hostnames</td></tr> <tr> <td>-p [c, t, l]</td><td>output order; default order is counts, time statistics, and lists</td></tr> <tr> <td>-m <i>filename</i></td><td>meta file(s)</td></tr> <tr> <td>-o <i>filename</i></td><td>output log file; default is stdout</td></tr> <tr> <td>-c [h, n, r, f, e, u, o, k, c, z]</td><td>count these items; default is: h, n, r, e, u, o, k, c</td></tr> <tr> <td></td><td> <ul style="list-style-type: none"> ■ 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 </td></tr> <tr> <td>-t [sx, mx, hx, xx, z]</td><td>find general statistics; default is: s5m5h24x10</td></tr> </table>	-i <i>filename</i>	input log file(s)	-P	proxy log format	-n <i>servername</i>	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 <i>filename</i>	meta file(s)	-o <i>filename</i>	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		<ul style="list-style-type: none"> ■ 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 	-t [sx, mx, hx, xx, z]	find general statistics; default is: s5m5h24x10
-i <i>filename</i>	input log file(s)																						
-P	proxy log format																						
-n <i>servername</i>	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 <i>filename</i>	meta file(s)																						
-o <i>filename</i>	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																						
	<ul style="list-style-type: none"> ■ 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 																						
-t [sx, mx, hx, xx, z]	find general statistics; default is: s5m5h24x10																						

flexanlg(1M)

-l [cx, hx]

- 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

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(1M), wsdeploy(1M)

get(1)

NAME	get – gets the values of the monitorable or configurable attributes.
SYNOPSIS	get [--monitor] --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>attributename</i> [<i>attribute_name</i>] *
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 <i>Sun Java System Application Server Administrator's Guide</i> 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	<i>attributename</i> attribute anme in the dotted notation.
EXAMPLES	EXAMPLE 1 Using get <pre> 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_1 server1.application.fortune.enabled=true server1.application.fortune.name=fortune server1.application.fortune.description=null server1.application.fortune.virtualServers=server1 </pre>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	set(1), reconfig(1), list(1)

help(1)

NAME	help – displays a list of all the commands available in the Command-line interface																																		
SYNOPSIS	asadmin --help or asadmin command --help																																		
DESCRIPTION	<p>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.</p> <p>The commands marked (<i>Enterprise Edition only</i>) are available in the Enterprise edition release of the Sun Java System Application Server. All the other commands are part of both the Standard and Enterprise editions of the Sun Java System Application Server.</p> <p>The following is a list of all the Command-line interface commands:</p> <table> <tr> <td>add-iiop-cluster-endpoint (<i>Enterprise Edition only</i>)</td><td>Adds an IIOP endpoint to the IIOP cluster.</td></tr> <tr> <td>add-resources</td><td>Adds one or more resources of type jdbc, jms, or javamail.</td></tr> <tr> <td>clear-session-store (<i>Enterprise Edition only</i>)</td><td>Clears the sessions for all application server instances from the HADB database.</td></tr> <tr> <td>configure-session-persistence (<i>Enterprise Edition only</i>)</td><td>Configures session persistence options for an application server instance.</td></tr> <tr> <td>create-acl</td><td>Creates an ACL (access control list).</td></tr> <tr> <td>create-authdb</td><td>Creates an authentication database.</td></tr> <tr> <td>create-auth-realm</td><td>Creates an authentication realm.</td></tr> <tr> <td>create-connection-pool</td><td>Creates a new connection group with the named group ID for the associated instance.</td></tr> <tr> <td>create-custom-resource</td><td>Creates a custom resource.</td></tr> <tr> <td>create-domain</td><td>Creates a domain.</td></tr> <tr> <td>create-file-user</td><td>Creates a file realm user in the keyfile.</td></tr> <tr> <td>create-http-listener</td><td>Creates an HTTP listener.</td></tr> <tr> <td>create-http-qos</td><td>Creates HTTP quality of service settings for the application server instance or virtual server.</td></tr> <tr> <td>create-iiop-listener</td><td>Creates an IIOP listener.</td></tr> <tr> <td>create-instance</td><td>Creates an application server instance.</td></tr> <tr> <td>create-javamail-resource</td><td>Creates a Java Mail resource.</td></tr> <tr> <td>create-jdbc-connection-pool</td><td>Creates a JDBC connection pool.</td></tr> </table>	add-iiop-cluster-endpoint (<i>Enterprise Edition only</i>)	Adds an IIOP endpoint to the IIOP cluster.	add-resources	Adds one or more resources of type jdbc, jms, or javamail.	clear-session-store (<i>Enterprise Edition only</i>)	Clears the sessions for all application server instances from the HADB database.	configure-session-persistence (<i>Enterprise Edition only</i>)	Configures session persistence options for an application server instance.	create-acl	Creates an ACL (access control list).	create-authdb	Creates an authentication database.	create-auth-realm	Creates an authentication realm.	create-connection-pool	Creates a new connection group with the named group ID for the associated instance.	create-custom-resource	Creates a custom resource.	create-domain	Creates a domain.	create-file-user	Creates a file realm user in the keyfile.	create-http-listener	Creates an HTTP listener.	create-http-qos	Creates HTTP quality of service settings for the application server instance or virtual server.	create-iiop-listener	Creates an IIOP listener.	create-instance	Creates an application server instance.	create-javamail-resource	Creates a Java Mail resource.	create-jdbc-connection-pool	Creates a JDBC connection pool.
add-iiop-cluster-endpoint (<i>Enterprise Edition only</i>)	Adds an IIOP endpoint to the IIOP cluster.																																		
add-resources	Adds one or more resources of type jdbc, jms, or javamail.																																		
clear-session-store (<i>Enterprise Edition only</i>)	Clears the sessions for all application server instances from the HADB database.																																		
configure-session-persistence (<i>Enterprise Edition only</i>)	Configures session persistence options for an application server instance.																																		
create-acl	Creates an ACL (access control list).																																		
create-authdb	Creates an authentication database.																																		
create-auth-realm	Creates an authentication realm.																																		
create-connection-pool	Creates a new connection group with the named group ID for the associated instance.																																		
create-custom-resource	Creates a custom resource.																																		
create-domain	Creates a domain.																																		
create-file-user	Creates a file realm user in the keyfile.																																		
create-http-listener	Creates an HTTP listener.																																		
create-http-qos	Creates HTTP quality of service settings for the application server instance or virtual server.																																		
create-iiop-listener	Creates an IIOP listener.																																		
create-instance	Creates an application server instance.																																		
create-javamail-resource	Creates a Java Mail resource.																																		
create-jdbc-connection-pool	Creates a JDBC connection pool.																																		

create-jdbc-resource	Creates a JDBC resource.
create-jmsdest	Creates a JMS (Java Message Service) destination.
create-jmsobj	Adds the named object.
create-jms-resource	Creates a JMS resource.
create-jndi-resource	Creates a JNDI resource.
create-jvm-options	Creates JVM options in java-config or profiler elements.
create-lifecycle-module	Creates a lifecycle module.
create-mime	Creates a MIME types file.
create-persistence-resource	Creates a persistence manager factory resource.
create-profiler	Creates a profiler for the JVM.
create-session-store (Enterprise Edition only)	Creates a session store for a cluster to store the session information.
create-ssl	Creates SSL settings for an HTTP listener, IIOP listener, or IIOP service.
create-virtual-server	Creates a virtual server.
delete-acl	Deletes an ACL.
delete-authdb	Deletes an authentication database.
delete-auth-realm	Deletes an authentication realm.
delete-connection-group	Deletes the connection group for the named instance.
delete-custom-resource	Deletes a custom resource.
delete-domain	Deletes a domain. This command can only be executed locally.
delete-file-user	Deletes a file realm user from the keyfile.
delete-http-listener	Deletes an HTTP listener.
delete-http-qos	Deletes HTTP quality of service settings for the application server instance or virtual server.
delete-iiop-cluster-endpoint (Enterprise Edition only)	Deletes IIOP endpoint from the IIOP cluster configuration.
delete-iiop-listener	Deletes an IIOP listener
delete-instance	Deletes an application server instance.
delete-javamail-resource	Deletes a Java Mail resource.

help(1)

delete-jdbc-connection-pool	Deletes a JDBC connection pool.
delete-jdbc-resource	Deletes a JDBC resource.
delete-jmsdest	Deletes a JMS destination.
delete-jmsobj	Destroys the named object.
delete-jms-resource	Deletes a JMS resource.
delete-jndi-resource	Deletes a JNDI resource.
delete-jvm-options	Deletes JVM options in java-config or profiler elements.
delete-lifecycle-module	Deletes a lifecycle module.
delete-mime	Deletes a MIME types file.
delete-persistence-resource	Deletes a persistence manager factory resource.
delete-profiler	Deletes a JVM profiler.
delete-ssl	Deletes SSL settings for an HTTP listener, IIOP listener, or IIOP service.
delete-virtual server	Deletes a virtual server.
deploy	Deploys an EJB, WEB, connector, appclient, or application component to the application server instance.
deploydir	Deploys an EJB, WEB, connector, appclient, or application component that is in the directory to the application server instance.
disable	Disables a deployed component in the application server instance.
display-license	Displays license information. This command can only be executed locally.
enable	Enables (allows to run) a deployed component in the application server instance.
export	Exports the value of an asadmin environment variable so that it can be used by the subsequent asadmin commands.
get	Gets the value of an attribute.
help	Displays help (description, usage, syntax, examples) for a given command, or general help for asadmin.
install-license	Installs the license file. This command can only be executed locally.
jms-ping	Pings the JMS provider to see if it is running.
list	Lists the configurable elements.

list-acls	Lists ACLs for an application server instance.
list-authdbs	Lists authentication databases.
list-auth-realms	Lists authentication realms.
list-components	Lists the deployed components for a server instance.
list-connection-groups	Gets the connection groups for the named instance.
list-custom-resources	Lists custom resources in a server instance
list-domains	Lists domains.
list-file-groups	Lists all the groups for a specified file realm user. If you do not specify a user, lists all groups for a server instance.
list-file-users	Lists all the file realm users in a server instance.
list-http-listeners	Lists HTTP listeners for a server instance.
list-instances	Lists application server instances in the domain.
list-iiop-cluster-config (Enterprise Edition only)	Lists the IIOP cluster configuration.
list-iiop-listeners	Lists IIOP listeners for a server instance.
list-javamail-resources	Lists Java Mail resources for a server instance.
list-jdbc-connection-pools	Lists JDBC connection pools for a server instance.
list-jdbc-resources	Lists JDBC resources for a server instance.
list-jmsdest	Lists JMS destinations for a server instance.
list-jmsobj	Gets all the named objects.
list-jms-resources	Lists JMS resources for a server instance
list-jndi-resources	Lists JNDI resources for a server instance.
list-lifecycle-modules	Lists lifecycle modules for a server instance.
list-mimes	Lists MIME types files for a server instance.
list-persistence-resources	Lists persistence manager factory resources for a server instance.
list-profilers	Lists JVM profilers for a server instance.
list-sub-components	Lists one or more EJBs or Servlets in a deployed module or in a module of the deployed application.
list-virtual-servers	Lists virtual servers for a server instance.
multimode	Allows you to execute multiple command while retaining your environment settings and remaining within asadmin.

help(1)

reconfig	Applies change to the server. Most changes do not take effect until they are applied.
restart-instance	Restarts the server instance.
set	Sets the value of an attribute.
show-component-status	Shows the status of a deployed component.
show-instance-status	Shows the status of a server instance (that is, whether it is running or not).
shutdown	Shuts down the Administration Server.
start-appserv	Starts the Administration Server and all the server instances. This command can only be executed locally.
start-domain	Starts all instances in the domain. This command can only be executed locally.
start-instance	Starts the server instance.
stop-appserv	Stops the Administration Server and all the server instances. This command can only be executed locally.
stop-domain	Stops all instances in the domain.
stop-instance	Stops the server instance.
undeploy	Removes the deployed component from the server instance.
unset	Unsets the exported environment variables for asadmin.
update-file-user	Updates an existing file realm user.
version	Displays version information for the Sun Java System Application Server.

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.

SEE ALSO asadmin(1M)

NAME	htpasswd – creates the user authentication files						
SYNOPSIS	htpasswd [-c] <i>passwdfile</i> <i>username</i>						
DESCRIPTION	Use the <code>htpasswd</code> utility to create the flat-files that store usernames and password for basic authentication of HTTP users. If <code>htpasswd</code> 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.						
OPTIONS	<table><tr><td><code>-c</code></td><td>creates the <code>passwdfile</code>. If the <code>passwdfile</code> already exists, it is rewritten and truncated.</td></tr><tr><td><i>passwdfile</i></td><td>name of the file to contain the username and password.</td></tr><tr><td><i>username</i></td><td>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></table>	<code>-c</code>	creates the <code>passwdfile</code> . If the <code>passwdfile</code> already exists, it is rewritten and truncated.	<i>passwdfile</i>	name of the file to contain the username and password.	<i>username</i>	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.
<code>-c</code>	creates the <code>passwdfile</code> . If the <code>passwdfile</code> already exists, it is rewritten and truncated.						
<i>passwdfile</i>	name of the file to contain the username and password.						
<i>username</i>	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.						
EXAMPLES	<code>htpasswd -c myauthen scott</code>						

install-license(1)

NAME	install-license – installs the license file
SYNOPSIS	install-license
DESCRIPTION	install-license prevents unauthorized use of the Sun Java System Application Server. Allows you to install the license file. This command can be run locally only.
EXAMPLES	EXAMPLE 1 Using install-license asadmin> install-license LICENSE agreement will be displayed. Do you agree with the terms of this license [YES NO] YES Enter license key> ***** Installed the license
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	display-license(1), version(1)

NAME	jms-ping – checks to see if the JMS provider is up and running
SYNOPSIS	jms-ping --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using jms-ping <pre>asadmin> jms-ping --user admin --password adminadmin --host bluestar --port 4848 server1</pre> JMS Ping Status=RUNNING
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-jmsdest(1), delete-jmsdest(1), list-jmsdest(1)

jspc(1M)

NAME	jspc – precompiles JSP source files into servlets	
SYNOPSIS	jspc [<i>options</i>] <i>jsp_files</i> or jspc [<i>options</i>] -webapp <i>dir</i>	
DESCRIPTION	<p>Use the <code>jspc</code> 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 <code>reload-interval</code> property to <code>-1</code> in the <code>jsp-config</code> element of the <code>sun-web.xml</code> file.</p> <p>For more information about the <code>sun-web.xml</code> file, see the <i>Sun Java System Application Server Developer's Guide</i>.</p>	
OPTIONS	<i>jsp_files</i>	one or more JSP files to be compiled.
	<code>-webapp</code> <i>dir</i>	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 <code>asadmin</code> <code>deploy</code> .
	<code>-q</code>	enables quiet mode (same as <code>-v0</code>). Only fatal error messages are displayed.
	<code>-d</code> <i>dir</i>	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 <code>jspc</code> is invoked.
	<code>-p</code> <i>name</i>	the name of the target package for all specified JSPs, overriding the default package generation performed by the <code>-d</code> option.
	<code>-c</code> <i>name</i>	the target class name of the first JSP compiled. Subsequent JSPs are unaffected.
	<code>-uribase</code> <i>dir</i>	the URI directory to which compilations are relative. Applies only to JSP files listed in the command, and not to JSP files specified with <code>-webapp</code> option. This is the location of each JSP file relative to the <code>uriroot</code> . If this cannot be determined, the default is <code>/</code> .
	<code>-uriroot</code> <i>dir</i>	the root directory against which URI files are resolved. Applies only to JSP files listed in the command, and not to JSP files specified with <code>-webapp</code> option. If this option is not specified, all parent directories of the first JSP page are searched for a <code>WEB-INF</code> subdirectory. The closest directory to the JSP page that has one is used. If none of the JSP's parent directories have a <code>WEB-INF</code> subdirectory, the directory from which <code>jspc</code> is invoked is used.
	<code>-genclass</code>	compiles the generated servlets into class files.

<code>-v [level]</code>	enables verbose mode. The level is optional; the default is 2. Possible level values are: <ul style="list-style-type: none"> ■ 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
<code>-mapped</code>	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.
<code>-die [code]</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.
<code>-webinc file</code>	creates partial servlet mappings for the <code>-webapp</code> option, which can be pasted into a <code>web.xml</code> file.
<code>-webxml file</code>	creates an entire <code>web.xml</code> file for the <code>-webapp</code> option.
<code>-ieplugin class_id</code>	specifies the Java plugin COM class ID for Internet Explorer. Used by the <code>jsp:plugin</code> 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(1M)`

list(1)

NAME	list – lists the configurable elements														
SYNOPSIS	list [--monitor] --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>element_name</i>														
DESCRIPTION	Lists the configurable elements (child nodes).														
OPTIONS	<table><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></table>	--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.
--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	<i>element_name</i> configurable or monitorable element name.														
EXAMPLES	<p>EXAMPLE 1 Using list for a server instance</p> <pre>asadmin> list --user admin --passwordfile passwords.txt --host localhost --port 4848 server1</pre> <p>List of configurable attributes for element server1</p> <pre>server1.jndi-resource server1.persistence-manager-factory-resource server1.application server1.http-service server1.connector-module server1.transaction-service server1.iiop-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 server1.iiop-service server1.jms-service server1.orb server1.resources</pre>														

EXAMPLE 1 Using list for a server instance *(Continued)*

```

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

SEE ALSO get(1), set(1), reconfig(1)

list-acls(1)

NAME	list-acls – gets the access control lists for the named instance
SYNOPSIS	list-acls --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
DESCRIPTION	Gets the access control lists associated with the named server instance.
OPTIONS	<p>--user administrative user associated for the instance.</p> <p>--password administrative password corresponding to the administrative user.</p> <p>--host host name of the machine hosting the administrative instance.</p> <p>--port administrative port number associated with the administrative host.</p> <p>--secure indicates communication with the administrative instance in secured mode.</p> <p>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</p>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-acls</p> <pre>asadmin> list-acls --user admin --password adminadmin --host fuyako --port 7070 server1 acl1 sampleACL</pre> <p>Where: <i>acl1</i> and <i>sampleACL</i> are the names of the ACLs listed.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	create-acl(1), delete-acl(1)

NAME	list-authdbs – gets the authorized database for the named instance
SYNOPSIS	<pre>list-authdbs --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--instance instance_name] --virtualserver virtualserver_ID authdb_ID</pre>
DESCRIPTION	Gets the access control lists associated with the named server instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-authdbs</p> <pre>asadmin> lsit-authdbs --user admin --password adminadmin --host fuyako --port 7070 --virtualserver server1 server1 default sampleAuth</pre> <p>Where: default and sampleAuth are the authdb IDs in virtual server server1 and instance server1 listed.</p> <pre>asadmin% list-authdbs --instance server1</pre>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-authdb(1), delete-authdb(1)

list-auth-realms(1)

NAME	list-auth-realms – lists the authorized realms associated with the named instance
SYNOPSIS	list-auth-realms --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-auth-realms <pre>asadmin> list-auth-realms --user admin --password adminadmin --host localhost --port 4848 server1 file ldap certificate db</pre> <p>Where file, ldap, certificate, and db are the auth realms listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-auth-realm(1), delete-auth-realm(1)

NAME	list-components – lists deployed J2EE components
SYNOPSIS	<pre>list-components --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--type application ejb web connector] instance_name</pre>
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	<pre>--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.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-components to list all components</p> <pre>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</pre> <p>Where: all the component that were deployed to the server1 instance are listed.</p> <p>EXAMPLE 2 Using list-components to list a web component</p> <pre>asadmin> list-components --user admin --passwordfile passwords.txt --port 4848 --host localhost --type web server1 simple web</pre> <p>Where: all the web component that was deployed to the server1 instance is listed.</p> <p>EXAMPLE 3 Using list-components to list an application component</p> <pre>asadmin> list-components --user admin --passwordfile passwords.txt --port 4848 --host localhost --type application server1 fortune application</pre>

list-components(1)

EXAMPLE 3 Using `list-components` to list an application component *(Continued)*

Where: all the application component that was deployed to the `server1` instance is listed.

EXIT STATUS

0	command executed successfully
1	error in executing the command

SEE ALSO `list-sub-components(1)`, `show-component-status(1)`

NAME	list-custom-resources – gets all the custom resources from the named instance
SYNOPSIS	list-custom-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-custom-resources <pre>asadmin> list-custom-resources --user admin --password adminadmin --host fuyako --port 7070 server1 sample_custom_resource</pre> <p>Where: <i>sample_custom_resource</i> is the custom resource listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-custom-resource(1), delete-custom-resource(1)

list-domains(1)

NAME	list-domains – lists all the domains														
SYNOPSIS	<pre>list-domains [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--local=false] [--passwordfile <i>filename</i>] [--secure -s]</pre>														
DESCRIPTION	<p>Use the <code>list-domains</code> command to list all the domains associated with the Sun Java System Application Server. The <code>list-domains</code> command can be run both locally and remotely. Set the <code>--local</code> 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.</p>														
OPTIONS	<table> <tr> <td><code>--user</code></td><td>administrative user associated for the instance.</td></tr> <tr> <td><code>--password</code></td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td><code>--host</code></td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td><code>--port</code></td><td>administrative port number associated with the administrative host.</td></tr> <tr> <td><code>--local</code></td><td>determines if the command should delegate the request to administrative instance or run locally.</td></tr> <tr> <td><code>--passwordfile</code></td><td>file containing passwords appropriate for the command (e.g., administrative instance).</td></tr> <tr> <td><code>--secure</code></td><td>if true, uses SSL/TLS to communicate with the administrative instance.</td></tr> </table>	<code>--user</code>	administrative user associated for the instance.	<code>--password</code>	administrative password corresponding to the administrative user.	<code>--host</code>	host name of the machine hosting the administrative instance.	<code>--port</code>	administrative port number associated with the administrative host.	<code>--local</code>	determines if the command should delegate the request to administrative instance or run locally.	<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).	<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.
<code>--user</code>	administrative user associated for the instance.														
<code>--password</code>	administrative password corresponding to the administrative user.														
<code>--host</code>	host name of the machine hosting the administrative instance.														
<code>--port</code>	administrative port number associated with the administrative host.														
<code>--local</code>	determines if the command should delegate the request to administrative instance or run locally.														
<code>--passwordfile</code>	file containing passwords appropriate for the command (e.g., administrative instance).														
<code>--secure</code>	if true, uses SSL/TLS to communicate with the administrative instance.														
EXAMPLES	<p>EXAMPLE 1 Using list-domains in local mode</p> <pre>asadmin> list-domains domain1 [/software/AS7SE/sep9/domains/domain1] domain2 [/u/mydomain/domain_root/domain2]</pre> <p>Where: the <code>domain1</code> and <code>domain2</code> are listed and their directory paths are identified.</p>														
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command										
0	command executed successfully														
1	error in executing the command														
SEE ALSO	<p><code>create-domain(1)</code>, <code>delete-domain(1)</code>, <code>start-domain(1)</code>, <code>stop-domain(1)</code>, <code>list-instances(1)</code>, <code>multimode(1)</code></p>														

NAME	list-file-groups – lists the file groups for the named instance
SYNOPSIS	<pre>list-file-groups --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--name username] instance_name</pre>
DESCRIPTION	Lists the available groups in the file user. If user_name option is not specified, then all groups will be listed.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using the list-file-groups command</p> <pre>asadmin> list-file-groups --user admin --password adminadmin --host fuyako --port 7070 --name sample_user server1 staff manager</pre> <p>Where: staff and manager are the groups for file user sample_user in instance server1.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	delete-file-user(1), update-file-user(1), create-file-user(1), list-file-users(1)

list-file-users(1)

NAME	list-file-users – lists the file users for the named instance
SYNOPSIS	list-file-users --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
DESCRIPTION	Lists all 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	<i>instance_name</i> 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	delete-file-user(1), update-file-user(1), create-file-user(1), list-file-groups(1)

NAME	list-http-listeners – gets the HTTP listeners for the named instance
SYNOPSIS	list-http-listeners --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile filename] [--secure -s] [--instance <i>instance_name</i>] httplistener_ID
DESCRIPTION	Gets 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.
EXAMPLES	EXAMPLE 1 Using list-http-listeners <pre>asadmin> list-http-listeners --user admin --password adminadmin --host fuyako --port 7070 --instance server1 sampleListener</pre> Deleted HTTP listener with id = sampleListener Where: sampleListener is the HTTP listener listed.
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-http-listener(1), delete-http-listener(1)

list-iiop-cluster-config(1)

NAME	list-iiop-cluster-config – lists the application server instances that are part of the IIOP cluster configuration
SYNOPSIS	list-iiop-cluster-config [--host <i>admin-host</i>] [--port <i>admin_port</i>] [--user <i>admin-username</i>] [--password <i>admin-password</i>] [--passwordfile <i>filename</i>] [--secure -s] <i>instance-name</i>
DESCRIPTION	<p>Lists the application server instances that are part of the IIOP cluster configuration. All the IIOP endpoints for server instances in the IIOP cluster can be obtained.</p> <p>The <code>list-iiop-cluster-config</code> command is available only in the <i>Enterprise Edition</i> of the Sun Java System Application Server.</p>
OPTIONS	<p>-H --host host name of the machine hosting the administrative instance.</p> <p>-p --port administrative port number associated with the administrative host.</p> <p>-u --user administrative user associated for the instance.</p> <p>-w --password administrative password corresponding to the administrative user.</p> <p>--passwordfile file containing passwords appropriate for the command (for example, administrative instance).</p> <p>-s --secure indicates communication with the administrative instance in secured mode.</p>
OPERANDS	<i>instance_name</i> name of the server instance at which this operation is targeted.
EXAMPLES	<p>EXAMPLE 1 List the IIOP cluster configuration for a particular server instance, server2</p> <pre>asadmin> list-iiop-cluster-config --user admin --password myPasswd server2</pre> <p>List of server instances and IIOP end points.</p>
SEE ALSO	add-iiop-cluster-endpoint(1), delete-iiop-cluster-endpoint(1)

NAME	list-iiop-listeners – gets the IIOP listeners for the named instance
SYNOPSIS	list-iiop-listeners --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-iiop-listeners <pre>asadmin> list-iiop-listeners --user admin --password adminadmin --host fuyako --port 7070 server1 orb-listener-1 sample_iiop_listener</pre> <p>Where: orb-listener-1 and sample_iiop_listener are the IIOP listeners listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-iiop-listener(1), delete-iiop-listener(1)

list-instances(1)

NAME	list-instances – lists all the instances in the server																
SYNOPSIS	list-instances [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--domain <i>domain_name</i>] [--local=false] [--passwordfile <i>filename</i>] [--secure -s]																
DESCRIPTION	Use the <code>list-instances</code> to list all the instance in the server. The <code>list-instances</code> 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	<table> <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>port number associated with the administrative host.</td></tr> <tr> <td>--domain</td><td>name of the domain.</td></tr> <tr> <td>--local</td><td>determines if the command should delegate the request to administrative instance or run locally.</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> </table>	--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.
--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	<p>EXAMPLE 1 Using list-instances in local mode</p> <pre>asadmin> list-instances --domain1 --local admin-server running server1 running</pre> <p>Where: the <code>server1</code> and <code>admin-server</code> instances for the <code>domain1</code> domain is listed.</p> <p>EXAMPLE 2 Using list-instances in remote mode</p> <pre>asadmin> list-instances --user admin --passwordfile passwords.txt --host localhost --port 4848 server1 [mayank:80] running</pre> <p>Where: the <code>server1</code> instance associated with the specified user, passwords, host, and port number specified is listed for the remote machine.</p>																
EXIT STATUS	<table> <tr> <td>0</td><td>command executed successfully</td></tr> <tr> <td>1</td><td>error in executing the command</td></tr> </table>	0	command executed successfully	1	error in executing the command												
0	command executed successfully																
1	error in executing the command																
SEE ALSO	<code>show-instance-status(1)</code>																

NAME	list-javamail-resources – gets all the Javamail resources from the named instance
SYNOPSIS	list-javamail-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-javamail-resources <pre>asadmin> list-javamail-resources --user admin --password adminadmin --host fuyako --port 7070 server1 sample_javamail_resource</pre> <p>Where: <i>sample_javamail_resource</i> is the Javamail resource listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-javamail-resource(1) delete-javamail-resource(1)

list-jdbc-connection-pools(1)

NAME	list-jdbc-connection-pools – gets all the JDBC connection pools from the named instance
SYNOPSIS	<pre>list-jdbc-connection-pools --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] instance_name</pre>
DESCRIPTION	Gets all the JDBC resources connection pools from the named instance.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-jdbc-connection-pools</p> <pre>asadmin> list-jdbc-connection-pools --user admin --password adminadmin --host fuyako --port 7070 server1 XA_connection_pool</pre> <p>Where: XA_connection_pool is the JDBC connection listed.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-jdbc-connection-pool(1), delete-jdbc-connection-pool(1)

NAME	list-jdbc-resources – gets all the JDBC resources from the named instance
SYNOPSIS	list-jdbc-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port <i>4848</i>] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using the list-jdbc-resources command</p> <pre>asadmin> list-jdbc-resources --user --password adminadmin --host fuyako --port 7070 server1 sample_jdbc_resource</pre> <p>Where: <i>sample_jdbc_resource</i> is the JDBC connection listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-jdbc-resource(1), delete-jdbc-resource(1)

list-jmsdest(1)

NAME	list-jmsdest – gets all the named destinations
SYNOPSIS	<pre>list-jmsdest --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--desttype type] instance_name</pre>
DESCRIPTION	Gets all the named destinations.
OPTIONS	<pre>--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. -destype type of JMS destination. Valid values are topic, and queue.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-jmsdest</p> <pre>asadmin> list-jmsdest ----user admin --password adminadmin --host bluestar --port 4848 server1 topic_dest topic {}</pre>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-jmsdest(1), delete-jmsdest(1)

NAME	list-jms-resources – gets all the JMS resources from the named instance
SYNOPSIS	<pre>list-jms-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--resourcetype type] instance_name</pre>
DESCRIPTION	Gets all the JMS resources from the named instance.
OPTIONS	<pre>--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 JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using the list-jms-resources command</p> <pre>asadmin> list-jms-resources --user admin --password adminadmin --host fuyako --port 7070 --resourcetype javax.jms.Queue server1 sample_jms_resource</pre> <p>Where: <i>sample_jms_resource</i> is the JMS resource listed.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-jms-resource(1), delete-jms-resource(1)

list-jndi-resources(1)

NAME	list-jndi-resources – gets all the JNDI resources from the named instance
SYNOPSIS	list-jndi-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 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
SEE ALSO	create-jndi-resource(1), delete-jndi-resource(1)

	list-lifecycle-modules(1)
NAME	list-lifecycle-modules – gets the lifecycle modules for the named instance
SYNOPSIS	list-lifecycle-modules --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>module_name</i>
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.
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-lifecycle-modules <pre>asadmin> list-lifecycle-modules --user admin --password adminadmin --host fuyako --port 7070 server1 customSetup</pre> <p>Where: customSetup is the lifecycle module listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-lifecycle-module(1), delete-lifecycle-module(1)

list-mimes(1)

NAME	list-mimes – gets the MIME types for the named instance
SYNOPSIS	<pre>list-mimes --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] instance_name</pre>
DESCRIPTION	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.
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-mimes</p> <pre>asadmin> list-mimes --user admin --password adminadmin --host fuyako --port 7070 server1 sampleMIME</pre> <p>Where: sampleMIME is the name of the MIME listed.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	create-mime(1), delete-mime(1)

list-persistence-resources(1)

NAME	list-persistence-resources – gets all the persistence resources from the named instance
SYNOPSIS	list-persistence-resources --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 using list-persistence-resources <pre>asadmin> list-persistence-resources --user admin --password adminadmin --host fuyako --port 7070 server1 sample_persistence_resource</pre> <p>Where: <i>sample_persistence_resource</i> is the persistence manager factory resource listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-persistence-resource(1), delete-persistence-resource(1)

list-profiler(1)

NAME	list-profiler – gets the profiler element in the named instance.
SYNOPSIS	list-profiler --user <i>user_name</i> --password <i>password</i> --host <i>hostname</i> --port <i>admin_port_number</i> [--instance <i>instance_name</i>]
DESCRIPTION	Gets the profiler element associated with the named server instance..
OPTIONS	--user identifies the user name associated with the named instance. --password identifies the password associated with the user name. --host identifies the host name for the machine. --port identifies the administrator port number associated with the hostname. --instance identifies the name of the instance associated with the JVM option to be created.
EXAMPLES	asadmin% list-profilers
SEE ALSO	create-profiler(1) delete-profiler(1)

NAME	list-profilers – lists the profiler elements in the named instance
SYNOPSIS	list-profilers --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using list-profilers <pre>asadmin> list-profilers --user admin --passwordfile passwords.txt --host localhost --port 4848 server1 sample_profiler</pre> <p>Where: sample_profiler is the profiler listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-profiler(1), delete-profiler(1)

list-sub-components(1)

NAME	list-sub-components – lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application
SYNOPSIS	<pre>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</pre>
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	<pre>--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.</pre>
OPERANDS	<i>module_name</i> name of the module containing the sub-components.
EXAMPLES	<p>EXAMPLE 1 Using list-sub-components</p> <pre>asadmin> list-sub-components --user admin --passwordfile passwords.txt --port 4848 --host localhost --instance server1 --type servlets --appname fortune fortune FortuneServlet Servlets</pre>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	deploy(1), deploydir(1), undeploy(1), enable(1), disable(1), list-components(1)

NAME	list-virtual-servers – gets the virtual servers in the named instance
SYNOPSIS	list-virtual-servers --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port 4848 [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	<p>EXAMPLE 1 Using list-virtual-servers</p> <pre>asadmin> list-virtual-servers --user admin --password adminadmin --host localhost --port 4848 server1 server1 sample_vs1</pre> <p>Where server1 and sample_vs1 are the virtual servers listed.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	create-virtual-server(1), delete-virtual-server(1)

multimode(1)

NAME	multimode – allows you to execute multiple commands while returning environment settings and remaining in the asadmin utility								
SYNOPSIS	multimode [--file <i>filename</i>] [--encoding <i>encode</i>] [--passwordfile <i>filename</i>] [--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 <i>multimode</i> , 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 <i>multimode</i> session; once you exit the second <i>multimode</i> environment, you return to your original <i>multimode</i> 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 <i>multimode</i> from a file. Commands in <i>multimode</i> , when piped from an input stream, and commands invoked from another program, cannot run in the interactive mode.								
OPTIONS	<table><tr><td>--file</td><td>consists of commands to be executed in <i>multimode</i>.</td></tr><tr><td>--encoding</td><td>system locale encoding method to be used.</td></tr><tr><td>--passwordfile</td><td>file containing the administrative passwords appropriate for the command (e.g., administrative instance)</td></tr><tr><td>--interactive</td><td>prompts you for the required options.</td></tr></table>	--file	consists of commands to be executed in <i>multimode</i> .	--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.
--file	consists of commands to be executed in <i>multimode</i> .								
--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	<p>EXAMPLE 1 Using multimode to execute multiple commands</p> <pre>example% asadmin multimode --file commands_file.txt</pre> <p>Where: example% is the system prompt. The multimode settings are executed from the <code>commands_file.txt</code> file.</p>								
EXIT STATUS	<table><tr><td>0</td><td>command executed successfully</td></tr><tr><td>1</td><td>error in executing the command</td></tr></table>	0	command executed successfully	1	error in executing the command				
0	command executed successfully								
1	error in executing the command								
SEE ALSO	export(1), unset(1)								

NAME	package-appclient – packs the application client container libraries and jar files
SYNOPSIS	package-appclient
DESCRIPTION	<p>Use the <code>package-appclient</code> command to pack the application client container libraries and jar files into an <code>appclient.jar</code> file. The created file is located at <code>appserver_install_dir/lib/appclient/appclient.jar</code>. The <code>appclient.jar</code> file provides an application client container package targeted at remote hosts that do not contain a server installation.</p> <p>The <code>appclient.jar</code> archive 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 <code>appclient.jar</code> produced on a Solaris SPARC platform cannot be used on a Windows client machine.</p> <p>After copying the <code>appclient.jar</code> file to a remote location, <code>unjar</code> it to get a set of libraries and jar files in the <code>appclient</code> directory</p> <p>After unjarring on the client machine, modify <code>appclient_install_dir/config/asenv.conf</code> (<code>asenv.bat</code> for Windows) as follows:</p> <ul style="list-style-type: none"> ■ set <code>AS_WEBSERVICES_LIB</code> to <code>appclient_install_dir/lib</code> ■ set <code>AS_NSS</code> to <code>appclient_install_dir/lib</code> (<code>appclient_install_dir\bin</code> for Windows) ■ set <code>AS_IMQ_LIB</code> to <code>appclient_install_dir/imq/lib</code> ■ set <code>AS_INSTALL</code> to <code>appclient_install_dir</code> ■ set <code>AS_JAVA</code> to your JDK 1.4 home directory ■ set <code>AS_ACC_CONFIG</code> to <code>appclient_install_dir/config/sun-acc.xml</code> <p>Modify <code>appclient_install_dir/config/sun-acc.xml</code> as follows:</p> <ul style="list-style-type: none"> ■ Ensure the <code>DOCTYPE</code> file references <code>appclient_install_dir/lib/dtds</code> ■ Ensure that <code>target-server</code> address attribute references the server machine. ■ Ensure that <code>target-server</code> port attribute references the ORB port on the remote machine. ■ Ensure that <code>log-service</code> references a log file; if the user wants to put log messages to a log file. <p>Modify <code>appclient_install_dir/bin/appclient</code> (<code>appclient.bat</code> for Windows) as follows:</p> <ul style="list-style-type: none"> ■ change token <code>%CONFIG_HOME%</code> to <code>appclient_install_dir/config</code>
ATTRIBUTES	See <code>attributes(5)</code> for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

package-appclient(1M)

SEE ALSO appclient (1M)

NAME	reconfig – applies the changes you have made for a server instance																
SYNOPSIS	<pre>reconfig --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure -s] [--discardmanualchanges=false] [--keepmanualchanges=false] instance_name</pre>																
DESCRIPTION	<p>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.</p> <p>The valid combinations are:</p> <ul style="list-style-type: none"> ■ --discardmanualchanges=true --keepmanualchanges=false this will discard your manual changes and use the admin changes. ■ --discardmanualchanges=false --keepmanualchanges=true this will keep your manual changes and discard the admin changes. ■ --discardmanualchanges=false --keepmanualchanges=false an error message is displayed if manual changes have been made. ■ --discardmanualchanges this will discard your manual changes and use the admin changes. ■ --keepmanualchanges this will keep your manual changes and discard the admin changes. ■ If no options are specified, an error message is displayed. <p>Use this command with discretion since there is no undo, and the changes applied are made directly to your config/backup/server.xml file.</p>																
OPTIONS	<table> <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> <tr> <td>--discardmanualchanges</td><td>defaults to false. When set to true, discards the changes made manually to the server.xml file.</td></tr> <tr> <td>--keepmanualchanges</td><td>defaults to false. When set to true, allows the manual changes made to the server.xml file to take affect.</td></tr> </table>	--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.
--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.																

reconfig(1)

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
```

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

SEE ALSO get(1), set(1), list(1)

NAME	restart-instance – restarts the specified server instance and all the services associated with it																
SYNOPSIS	<pre>restart-instance [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename] [--secure -s] instance_name</pre>																
DESCRIPTION	<p>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.</p>																
OPTIONS	<table> <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>--local</td><td>determines if the command should delegate the request to administrative instance or run locally.</td></tr> <tr> <td>--domain</td><td>name of the domain.</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> </table>	--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.
--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	<i>instance_name</i> name of the instance to be restarted.																
EXAMPLES	<p>EXAMPLE 1 Using restart-instance in local mode</p> <pre>asadmin> restart-instance --local --domain domain1 server1</pre> <p>Instance server1 started</p> <p>Where: server1 is the name of the instance restarted on the domain1 domain.</p> <p>EXAMPLE 2 Using restart-instance in remote mode</p> <pre>asadmin> restart-instance --user admin --password adminadmin --host bluestar --port 4848 server1</pre> <p>Instance server1 started</p> <p>Where: server1 is the name of the instance restarted. The restarted instance is associated with the user, password, host, and port number specified.</p>																

restart-instance(1)

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

SEE ALSO	delete-instance(1), start-instance(1), create-instance(1), stop-instance(1), start-appserv(1), stop-appserv(1), start-domain(1), stop-domain(1)
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------

set(1)

NAME	set – sets the values of attributes
SYNOPSIS	set [--monitor] --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] attributename=value [<i>attribute_name=value</i>] *
DESCRIPTION	Sets the values of one or more configurable attribute. The settings do not take affect until you run the <code>reconfig</code> command.
OPTIONS	<p>--monitor defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.</p> <p>--user administrative user associated for the instance.</p> <p>--password administrative password corresponding to the administrative user.</p> <p>--host host name of the machine hosting the administrative instance.</p> <p>--port administrative port number associated with the administrative host.</p> <p>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</p> <p>--secure if true, uses SSL/TLS to communicate with the administrative instance.</p>
OPERANDS	attributename=value identifies the configurable or monitorable attribute name and its value.
EXAMPLES	<p>EXAMPLE 1 Using set</p> <pre>asadmin> set --user admin --passwordfile passwords.txt --host localhost --port 4848 server1.application.fortune.enabled=false Attribute enabled set to false</pre>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	get(1), reconfig(1), list(1)

show-component-status(1)

NAME	show-component-status – displays the status of the deployed component
SYNOPSIS	<pre>show-component-status --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] <i>component_name</i></pre>
DESCRIPTION	Use the show-component-status command to get the status of 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	<pre>--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.</pre>
OPERANDS	<i>component_name</i> name of the component to be listed.
EXAMPLES	<p>EXAMPLE 1 Using show-component-status to show an application</p> <pre>asadmin> show-component-status --user admin --passwordfile passwords.txt --host bluestar --port 4848 fortune Status of application fortune is enabled</pre> <p>Where: the status of the fortune application is shown.</p> <p>EXAMPLE 2 Using show-component-status to show a WAR module</p> <pre>asadmin> show-component-status --user admin --passwordfile passwords.txt --host bluestar --port 4848 simple Status of WAR module simple is enabled</pre> <p>Where: the status of the simple WAR module is shown.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	list-components(1), list-sub-components(1)

NAME	show-instance-status – displays the status of the server instance specified.
SYNOPSIS	show-instance-status --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] --port <i>4848</i> [--local=false] [--passwordfile <i>filename</i>] --secure -s <i>instance_name</i>
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	<i>instance_name</i> name of the instance.
EXAMPLES	EXAMPLE 1 Using show-instance-status <pre>asadmin> show-instance-status --user admin --password adminadmin --host localhost --port 4848 server1 Status of instance server1 is running</pre> <p>Where:the status of the <i>server1</i> instance is shown.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	list-instances(1)

shutdown(1)

NAME	shutdown – brings down the administration server
SYNOPSIS	shutdown [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s]
DESCRIPTION	shutdown gracefully brings down the administration server. 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 adminadmin --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
SEE ALSO	start-instance(1), stop-instance(1), restart-instance(1), start-domain(1), stop-domain(1), start-appserv(1), stop-appserv(1)

NAME	start-appserv – attempts to start all the administrative domains defined for an installation of the application server
SYNOPSIS	start-appserv
DESCRIPTION	<p>Starts the administrative server and all the application server instances of each administrative domain defined for a particular installation of the application server. This command will only start the administrative domains and application server instances if the user executing the command has execute privileges for the administrative server and application server instance startup scripts.</p> <p>Use the <code>start-domain</code> and <code>stop-domain</code> commands to start and stop specific administrative domains.</p> <p>The <code>start-appserv</code> command can be run locally only. One or more domain must already exist.</p>
EXAMPLES	<p>EXAMPLE 1 Using start-appserv</p> <pre>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</pre> <p>Where: the <code>admin-server</code> and <code>server1</code> instances are started along with the domain <code>domain1</code> they are associated with. The <code>admin-server</code> instance and the <code>sample-domain</code> domain it is associated with are also started.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	<code>create-instance(1)</code> , <code>delete-instance(1)</code> , <code>start-instance(1)</code> , <code>stop-instance(1)</code> , <code>restart-instance(1)</code> , <code>stop-appserv(1)</code> , <code>start-domain(1)</code> , <code>stop-domain(1)</code>

start-domain(1)

NAME	start-domain – starts the given domain
SYNOPSIS	start-domain [--domain <i>domain_name</i>]
DESCRIPTION	Use the <code>start-domain</code> command to start all the instances in the specified domain. If the <code>--domain</code> option is not specified, and there is only one domain, all the instances in that domain are started. The <code>start-domain</code> command can be run locally only. The domain must currently exist on the local machine
OPTIONS	<code>--domain</code> name of the domain; must be a unique name.
EXAMPLES	<p>EXAMPLE 1 Using start-domain</p> <pre>asadmin> start-domain --domain domain1 instance domain1:admin-server started instance domain1:server1 started domain domain1 started</pre> <p>Where: the domain1 domain is started. By starting the domain, the admin-server and server1 instances in the domain are also started.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	<code>create-domain(1)</code> , <code>delete-domain(1)</code> , <code>stop-domain(1)</code> , <code>list-domains(1)</code> , <code>start-appserv(1)</code> , <code>stop-appserv(1)</code> , <code>start-instance(1)</code> , <code>stop-instance(1)</code>

NAME	start-instance – starts a server instance and all the services associated with it																		
SYNOPSIS	start-instance [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>local_host</i>] [--port 4848] [--local=false] [--domain <i>domain_name</i>] [--debug=false] [--passwordfile <i>filename</i>] [--secure -s] <i>instance_name</i>																		
DESCRIPTION	<p>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.</p>																		
OPTIONS	<table> <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>--local</td><td>determines if the command should delegate the request to administrative instance or run locally.</td></tr> <tr> <td>--domain</td><td>name of the domain.</td></tr> <tr> <td>--debug</td><td>starts the instance in debug mode.</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> </table>	--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.
--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	<i>instance_name</i> name of the instance to be started.																		
EXAMPLES	<p>EXAMPLE 1 Using start-instance in local mode</p> <pre>asadmin> start-instance --domain domain1 --local=true admin-server</pre> <p>Instance admin-server started</p> <p>Where: the admin-server instance is started on the local domain1 domain.</p> <p>EXAMPLE 2 Using start-instance in remote mode</p> <pre>asadmin> start-instance --user admin --password bluestar --host localhost --port 4848 server1</pre> <p>Instance server1 started</p> <p>Where: the server1 instance is started on the remote domain associated with the specified user, password, host, and port number.</p>																		

start-instance(1)

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

SEE ALSO	delete-instance(1), create-instance(1), stop-instance(1), restart-instance(1), start-appserv(1), stop-appserv(1), start-domain(1), stop-domain(1)
-----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------

NAME	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	<p>EXAMPLE 1 Using stop-appserv</p> <pre>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</pre> <p>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.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	create-instance(1), delete-instance(1), start-instance(1), stop-instance(1), restart-instance(1), start-appserv(1), start-domain(1), stop-domain(1)

stop-domain(1)

NAME	stop-domain – stops the given domain
SYNOPSIS	<pre>stop-domain [--user admin_user] [--password admin_password] [--host localhost] [--port 4848] [--local=false] [--domain domain_name] [--adminserv=true] [--passwordfile filename] [--secure -s]</pre>
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	<p>--user administrative user associated for the instance.</p> <p>--password administrative password corresponding to the administrative user.</p> <p>--host host name of the machine hosting the administrative instance.</p> <p>--port administrative port number associated with the administrative host.</p> <p>--local determines if the command should delegate the request to administrative instance or run locally.</p> <p>--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.</p> <p>--adminserv determines if the administrative instance should be stopped along with other instances.</p> <p>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</p> <p>--secure if set to true, uses SSL/TLS to communicate with the administrative instance.</p>
EXAMPLES	<p>EXAMPLE 1 Using stop-domain in local mode</p> <pre>asadmin> stop-domain --domain domain1 --adminserv=true Instance domain1:admin-server stopped Instance domain1:server1 stopped Domain domain1 stopped</pre> <p>Where: the domain1 domain is stopped. By stopping the domain the admin-server and server1 instance in the domain are also stopped.</p> <p>EXAMPLE 2 Using stop-domain in remote mode</p> <pre>asadmin> stop-domain --user admin --passwordfile passwords.txt --host bluestar --port 6886 Domain stopped remotely</pre> <p>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.</p>

stop-domain(1)

EXIT STATUS

0	command executed successfully
1	error in executing the command

SEE ALSO create-domain(1), delete-domain(1), start-domain(1), list-domains(1), start-appserv(1), stop-appserv(1), start-instance(1), stop-instance(1), multimode(1)

stop-instance(1)

NAME	stop-instance – stops the specified server instance and all the services associated with it
SYNOPSIS	stop-instance [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>local_host</i>] [--port 4848] [--local=false] [--domain <i>domain_name</i>] [--secure -s] <i>instance_name</i>
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	<i>instance_name</i> name of the instance to be stopped.
EXAMPLES	<p>EXAMPLE 1 Using stop-instance in local mode</p> <pre>asadmin> stop-instance --local --domain domain1 server1</pre> <p>Instance server1 stopped</p> <p>Where: the server1 instance associated with the domain1 domain is stopped locally.</p> <p>EXAMPLE 2 Using stop-instance in remote mode</p> <pre>asadmin> stop-instance --user admin --password bluestar --host localhost --port 4848 server1</pre> <p>Instance server1 stopped</p> <p>Where: the server1 instance associated with the named user, password, host and port is deleted from the remote machine.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command

stop-instance(1)

SEE ALSO delete-instance(1), start-instance(1), create-instance(1),
restart-instance(1), start-appserv(1), stop-appserv(1), start-domain(1),
stop-domain(1)

undeploy(1)

NAME	undeploy – removes the component from the named instance.
SYNOPSIS	undeploy --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--type <i>application ejb web connector</i>] [--instance <i>instance_name</i>] <i>component_name</i>
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	<p>--user administrative user associated for the instance.</p> <p>--password administrative password corresponding to the administrative user.</p> <p>--host host name of the machine hosting the administrative instance.</p> <p>--port administrative port number associated with the administrative host.</p> <p>--passwordfile file containing passwords appropriate for the command (e.g., administrative instance).</p> <p>--secure if true, uses SSL/TLS to communicate with the administrative instance.</p> <p>--type identifies the type of component to be deployed; defaults to the type application.</p> <p>--instance name of the instance.</p>
OPERANDS	<i>component_name</i> name of the deployable component.
EXAMPLES	<p>EXAMPLE 1 Using undeploy</p> <pre>asadmin> undeploy --user admin --password adminadmin --host localhost --port 4848 --type application --instance server1 fortune Undeployed the application:fortune</pre> <p>Where: the fortune application is undeployed from the server1 instance.</p>
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>
SEE ALSO	deploy(1), deploydir(1), enable(1), disable(1), list-components(1)

unset(1)

NAME	unset – removes one or more variables from the environment.
SYNOPSIS	unset <i>env_var</i> [<i>env_var</i>] *
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	<i>env_var</i> environment variable to be removed.
EXAMPLES	<p>EXAMPLE 1 Using unset to remove environment variables</p> <pre>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=*****</pre> <p>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.</p>
EXIT STATUS	0 command executed successfully 1 error in executing the command
SEE ALSO	export(1), multimode(1)

update-file-user(1)

NAME	update-file-user – updates a current file user as specified
SYNOPSIS	<pre>update-file-user --user <i>admin_user</i> [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--passwordfile <i>filename</i>] [--secure -s] [--instance <i>instance_name</i>] [--userpassword <i>user_password</i>] [--groups <i>user_groups:[user_groups]*</i>] <i>user_name</i></pre>
DESCRIPTION	Updates an existing entry in keyfile by the specified <i>user_name</i> , <i>user_password</i> and groups. Multiple groups can be entered by separating them, with a colon, ":".
OPTIONS	<pre>--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.</pre>
OPERANDS	<i>user_name</i> name of file user.
EXAMPLES	<p>EXAMPLE 1 Using the update-file-user command to update a file user</p> <pre>asadmin> update-file-user --user admin --password adminadmin --host fuyako --port 7070 --instance server1 --userpassword sample_password --groups staff:manager:engineer sample_user Updated File user sample_user</pre> <p>Where: the <i>sample_user</i> is the file user updated with the updated user password and groups.</p>
EXIT STATUS	<pre>0 command executed successfully 1 error in executing the command</pre>
SEE ALSO	delete-file-user(1), list-file-users(1), create-file-user(1), list-file-groups(1)

NAME	verifier – validates the J2EE Deployment Descriptors against application server DTDs
SYNOPSIS	verifier [-v] [-d <i>destination_directory</i>] [-r [a w f]] <i>jar_filename</i>
DESCRIPTION	<p>Use the <i>verifier</i> utility to validate the J2EE deployment descriptors and the Application Server specific deployment descriptors. If the application is not J2EE compliant, an error message is printed.</p> <p>When you run the <i>verifier</i> 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 <i>jar_filename_verified.xml</i> and <i>jar_filename_verified.txt</i>.</p> <p>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 <i>static-verification</i> which may contain the tags <i>application</i>, <i>ejb</i>, <i>web</i>, <i>appclient</i>, <i>connector</i>, <i>other</i>, <i>error</i> and <i>failure-count</i>. 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 <i>application</i>, <i>ejb</i>, <i>web</i>, <i>other</i>, and <i>failure-count</i>.</p> <p>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.</p>
OPTIONS	<p>-v verbose debugging is turned on.</p> <p>-d identifies where the result files get placed.</p> <p>-r identifies the reporting level defined as one of the following:</p> <ul style="list-style-type: none"> ■ 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 <p><i>jar_filename</i> name of the ear/war/jar file to perform static verification on. The results of verification are placed in two files <i>jar_filename_verified.xml</i> and <i>jar_filename_verified.txt</i> in the destination directory.</p>
EXAMPLES	<p>EXAMPLE 1 Using verifier in the Verbose Mode</p> <pre>example% verifier -v -d /verifier-results -rf sample.ear</pre> <p>Where -v runs the verifier in verbose mode, -d specifies the destination directory, and -rf displays only the failures. The results are stored in <i>/verifier-results/sample.ear_verified.xml</i> and <i>/verifier-results/sample.ear_verified.txt</i>.</p>
SEE ALSO	asadmin(1M)

version(1)

NAME	version – displays the version information																				
SYNOPSIS	<pre> version [--user <i>admin_user</i>] [--password <i>admin_password</i>] [--host <i>localhost</i>] [--port 4848] [--secure -s] [--passwordfile <i>filename</i>] [--terse=false] [--echo=false] [--interactive] [--verbose=false] </pre>																				
DESCRIPTION	version displays the version information. This command is supported in remote mode only.																				
OPTIONS	<table> <tr> <td>-u --user</td><td>administrative user associated for the instance.</td></tr> <tr> <td>-w --password</td><td>administrative password corresponding to the administrative user.</td></tr> <tr> <td>-H --host</td><td>host name of the machine hosting the administrative instance.</td></tr> <tr> <td>-p --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>-s --secure</td><td>indicates communication with the administrative instance in secured mode.</td></tr> <tr> <td>--terse</td><td>indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script.</td></tr> <tr> <td>--echo</td><td>setting to true will echo the command line statement on the standard output.</td></tr> <tr> <td>--interactive</td><td>prompts you for the required options that are not already specified.</td></tr> <tr> <td>--verbose</td><td>displays version information in detail.</td></tr> </table>	-u --user	administrative user associated for the instance.	-w --password	administrative password corresponding to the administrative user.	-H --host	host name of the machine hosting the administrative instance.	-p --port	administrative port number associated with the administrative host.	--passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).	-s --secure	indicates communication with the administrative instance in secured mode.	--terse	indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script.	--echo	setting to true will echo the command line statement on the standard output.	--interactive	prompts you for the required options that are not already specified.	--verbose	displays version information in detail.
-u --user	administrative user associated for the instance.																				
-w --password	administrative password corresponding to the administrative user.																				
-H --host	host name of the machine hosting the administrative instance.																				
-p --port	administrative port number associated with the administrative host.																				
--passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).																				
-s --secure	indicates communication with the administrative instance in secured mode.																				
--terse	indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script.																				
--echo	setting to true will echo the command line statement on the standard output.																				
--interactive	prompts you for the required options that are not already specified.																				
--verbose	displays version information in detail.																				
EXAMPLES	<p>EXAMPLE 1 Using remote mode to display version</p> <pre> asadmin> version Sun Java System Application Server 7 2004Q2 </pre> <p>EXAMPLE 2 Using remote mode to display version in detail</p> <pre> asadmin> export --user admin --password adminadmin --host bluestar --port 4848 --verbose Sun Java System Application Server 7 2004Q2 (build A021930-126949) </pre>																				
EXIT STATUS	<p>0 command executed successfully</p> <p>1 error in executing the command</p>																				
SEE ALSO	help(1)																				

NAME	wscompile – generates stubs, ties, serializers, and WSDL files used in JAX-RPC clients and services	
SYNOPSIS	wscompile [<i>options</i>] <i>configuration_file</i>	
DESCRIPTION	<p>Use the <code>wscompile</code> command to generate the client stubs and server-side ties for the service definition interface that represents the web service interface. Additionally, use the <code>wscompile</code> command to generate the WSDL description of the web service interface which is then used to generate the implementation artifacts.</p> <p>In addition to supporting the generation of stubs, ties, server configuration, and WSDL documents from a set of RMI interfaces, <code>wscompile</code> also supports generating stubs, ties and remote interfaces from a WSDL document.</p> <p>You must specify one of the <code>-gen</code> options in order to use <code>wscompile</code> as a stand alone generator. You must use either <code>-import</code> (for WSDL) or <code>-define</code> (for an RMI interface) along with the <code>-model</code> option in order to use <code>wscompile</code> in conjunction with <code>wsdeploy</code>.</p> <p>Invoking the <code>wscompile</code> command without specifying any arguments outputs the usage information.</p>	
OPTIONS	<ul style="list-style-type: none"> <code>-cp path</code> location of the input class files. <code>-classpath path</code> same as <code>-cp path</code> option. <code>-d directory</code> where to place the generated output files. <code>-define</code> read the service's RMI interface, define a service. Use this option with the <code>-model</code> option in order to create a model file for use with the <code>wsdeploy</code> command. <code>-f:features</code> enables the given features. Features are specified as a comma separated list of features. See the list of supported features below. <code>-features:features</code> same as <code>-f:features</code> option. <code>-g</code> generates the debugging information. <code>-gen</code> generates the client-side artifacts. <code>-gen:client</code> same as <code>-gen</code> option. <code>-gen:server</code> generates the server-side artifacts. <code>-gen:both</code> generates client and server artifacts. <code>-httpproxy:host:port</code> specifies a HTTP proxy server; defaults to port 8080. <code>-import</code> 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 <code>-model</code> option in order to create a model file for use with the <code>wsdeploy</code> command. 	

wscompile(1M)

	<p>-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.</p> <p>-keep keeps the generated files.</p> <p>-nd <i>directory</i> directory for the non-class generated files.</p> <p>-O optimizes the generated code.</p> <p>-s <i>directory</i> directory for the generated source files.</p> <p>-verbose outputs messages about what the compiler is doing.</p> <p>-version prints version information.</p> <p>Exactly one of the -input, -define, -gen options must be specified.</p>
SUPPORTED FEATURES	<p>datahandleronly always map attachments to data handler type</p> <p>explicitcontext turn on explicit service context mapping.</p> <p>infix=<i>name</i> specify an infix to use for generated serializers.</p> <p>nodatabinding turn off data binding for literal encoding.</p> <p>noencodedtypes turn off encoding type information.</p> <p>nomultirefs turn off support for multiple references.</p> <p>novalidation turn off validation for the imported WSDL file.</p> <p>searchschema search schema aggressively for subtypes.</p> <p>serializeinterfaces turn on direct serialization of interface types.</p> <p>Note: the -gen options are not compatible with wsdeploy.</p>
CONFIGURATION FILE FORMAT	<p>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:</p> <pre><?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></pre> <p>The configuration element may contain exactly one <service>, <wsdl> or <modelfile>.</p>

**SERVICE
ELEMENT**

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:

```
<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>
```

WSDL ELEMENT

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:

```
<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:

```
<modelfile location="mymodel.Z"/>
```

EXAMPLES

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(1M)

SEE ALSO wsdeploy(1M)

NAME	wsdeploy – reads a WAR file and the jaxrpc-ri.xml file and generates another WAR file that is ready for deployment
SYNOPSIS	wsdeploy [<i>options</i>] <i>input-WAR-file</i>
DESCRIPTION	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	<ul style="list-style-type: none"> -classpath <i>path</i> location of the input class files. -keep keep temporary files. -tmpdir temporary directory to use. -o <i>output WAR file</i> required; location of the generated WAR file. -verbose outputs messages about what the compiler is doing. -version prints version information.
EXAMPLES	<p>The input WAR file for wsdeploy will typically have the following structure:</p> <pre> 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 </pre> <p>Running wsdeploy on the above WAR file results in the following actions:</p> <pre> 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 </pre> <p>EXAMPLE 1 Creating the output WAR file</p> <pre>wsdeploy -o Hello.war Input.war</pre> <p>Where the deployable WAR file Hello.war is generated.</p>

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 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(1M)

EXAMPLE 1 Creating the output WAR file *(Continued)*

```
endpointName="MyHello"  
urlPattern="/hello"/>  
</webServices>
```

SEE ALSO wscompile(1M)

wsdeploy(1M)

Index

A

add-iiop-cluster-endpoint — adds an application server instance with its IIOP endpoint to the IIOP cluster, 12

add-resources — registers the named resource in the XML file specified., 14

adds a lifecycle module for the named instance — create-lifecycle-module, 85

adds a new access control list file for the named instance — create-acl, 55

adds a new HTTP listener socket — create-http-listener, 64

adds the IIOP listener for the named instance — create-iiop-listener, 68

adds the named destination — create-jmsdest, 78

adds the new authorized database for the named instance — create-authdb, 56

adds an application server instance with its IIOP endpoint to the IIOP cluster — add-iiop-cluster-endpoint, 12

deletes an IIOP server instance or an IIOP endpoint from the IIOP cluster configuration — delete-iiop-cluster-endpoint, 105

allows you to execute multiple subcommands while returning environment settings and remaining in the asadmin utility — multimode, 174

analyzes access log files — flexanlg, 133

appclient — launches the Application Client Container and invokes the client application packaged in the application JAR file, 15

applies the changes you have made for a server instance — reconfig, 177

asadmin — utility for performing administrative tasks for the Sun ONE Application Server, 17

asant — launches the Jakarta Ant tool, 38

automates the cluster setup process in a typical single machine configuration — clsetup, 46

B

brings down the administration server and associated instances — shutdown, 184

C

capture-schema — stores the database metadata (schema) in a file for use in mapping and execution, 40

checks to see if the JMS provider is up and running — jms-ping, 143

clear-session-store — clears all the sessions from the persistent store, 45

clears all the sessions from the persistent store — clear-session-store, 45

clsetup — automates the cluster setup process in a typical single machine configuration, 46

configure-session-persistence — enables configuration of parameters related to session persistence, 52

create-acl — adds a new access control list file for the named instance, 55
 create-authdb — adds the new authorized database for the named instance, 56
 create-domain — creates a domain with the given name, 61
 create-file-user — creates a new file user, 62
 create-http-listener — adds a new HTTP listener socket, 64
 create-iiop-listener — adds the IIOP listener for the named instance, 68
 create-instance — creates an application server instance with the specified instance name, 70
 create-javamail-resource — registers the Javamail resource to the named instance, 72
 create-jdbc-connection-pool — registers the JDBC connection pool to the named instance, 74
 create-jdbc-resource — registers the JDBC resource to the named instance, 77
 create-jms-resource — registers the JMS resource to the named instance, 79
 create-jmsdest — adds the named destination, 78
 create-jndi-resource — registers the JNDI resource to the named instance, 81
 create-jvm-options — creates the JVM options from the Java configuration or profiler elements, 83
 create-lifecycle-module — adds a lifecycle module for the named instance, 85
 create-persistence-resource — registers the persistence resource to the named instance, 88
 create-ssl — Creates the SSL element in the HTTP listener or IIOP listener, 93
 creates a domain with the given name — create-domain, 61
 creates a new file user — create-file-user, 62
 creates an application server instance with the specified instance name — create-instance, 70
 creates the JVM options from the Java configuration or profiler elements — create-jvm-options, 83
 Creates the SSL element in the HTTP listener or IIOP listener — create-ssl, 93

creates the user authentication files — htpasswd, 141

D

delete-acl — removes the access control list file for the named instance, 97
 delete-authdb — removes the authorized database for the named instance, 98
 delete-custom-resource — removes the custom resource from the named instance, 100
 delete-domain — deletes the given domain, 101
 delete-file-user — removes the named file user, 102
 delete-http-listener — removes the HTTP listener for the named instance, 103
 delete-http-qos — removes the quality of service parameter for the named instance, 104
 delete-iiop-listener — removes the IIOP listener for the named instance, 107
 delete-instance — deletes the instance that is not running, 108
 delete-javamail-resource — removes the Javamail resource from the named instance, 110
 delete-jdbc-connection-pool — removes the JDBC connection pool from the named instance, 111
 delete-jdbc-resource — removes the JDBC resource from the named instance, 112
 delete-jms-resource — removes the JMS resource from the named instance, 114
 delete-jmsdest — destroys the named destination, 113
 delete-jvm-options — deletes the JVM options from the Java configuration or profiler elements, 116
 delete-lifecycle-module — removes the lifecycle module for the named instance, 118
 delete-mime — removes the MIME type for the named instance, 119
 delete-persistence-resource — removes the persistence resource from the named instance, 120

- delete-ssl — deletes the ssl element from the HTTP listener or IIOP listener, 122
- delete-virtual-server — deletes the virtual server with the named virtual server ID, 123
- deletes the given domain — delete-domain, 101
- deletes the instance that is not running. — delete-instance, 108
- deletes the JVM options from the Java configuration or profiler elements — delete-jvm-options, 116
- deletes the ssl element from the HTTP listener or IIOP listener — delete-ssl, 122
- deletes the virtual server with the named virtual server ID — delete-virtual-server, 123
- deploy — deploys the specified component, 124
- deploydir — deploys the J2EE component that is in the directory located on the server machine, 127
- deploys the J2EE component that is in the directory located on the server machine — deploydir, 127
- deploys the specified component — deploy, 124
- destroys the named destination — delete-jmsdest, 113
- disable — stops the specified component, 129
- display-license — displays the license information, 130
- displays a list of all the commands available in the Command-line interface — help, 136
- displays the license information — display-license, 130
- displays the status of the deployed component — show-component-status, 182
- displays the status of the server instance specified. — show-instance-status, 183

E

- enable — runs the specified component, 131
- enables configuration of parameters related to session persistence — configure-session-persistence, 52

- export — marks a variable name for automatic export to the environment of subsequent commands in multimode, 132
- version — displays the version information, 198

F

- flexanlg — analyzes access log files, 133

G

- gets all the custom resources from the named instance — list-custom-resources, 153
- gets all the Javamail resources from the named instance — list-javamail-resources, 161
- gets all the JDBC connection pools from the named instance — list-jdbc-connection-pools, 162
- gets all the JNDI resources from the named instance — list-jndi-resources, 166
- gets all the named destinations — list-jmsdest, 164
- gets all the persistence resources from the named instance — list-persistence-resources, 169
- gets the access control lists for the named instance — list-acls, 148
- gets the HTTP listeners for the named instance — list-http-listeners, 157
- gets the IIOP listeners for the named instance — list-iiop-listeners, 159
- gets the lifecycle modules for the named instance — list-lifecycle-modules, 167
- gets the MIME types for the named instance — list-mimes, 168
- gets the virtual servers in the named instance — list-virtual-servers, 173

H

- delete-iiop-cluster-endpoint — deletes an IIOP server instance or an IIOP endpoint from the IIOP cluster configuration, 105

help — displays a list of all the commands available in the Command-line interface, 136
htpasswd — creates the user authentication files, 141

I

install-license — installs the license file, 142
installs the license file — install-license, 142

J

jms-ping — checks to see if the JMS provider is up and running, 143
jspc — precompiles JSP source files into servlets, 144

L

launches the Application Client Container and invokes the client application packaged in the application JAR file. — appclient, 15
launches the Jakarta Ant tool — asant, 38
list — lists the configurable elements, 146
list-acls — gets the access control lists for the named instance, 148
list-components — Lists deployed J2EE components, 151
list-custom-resources — gets all the custom resources from the named instance, 153
list-domains — lists all the domains, 154
list-file-groups — lists the file groups for the named instance, 155
list-file-users — lists the file users for the named instance, 156
list-http-listeners — gets the HTTP listeners for the named instance, 157
list-iiop-cluster-config — lists the server instances that are part of the IIOP cluster configuration, 158
list-iiop-listeners — gets the IIOP listeners for the named instance, 159
list-javamail-resources — gets all the Javamail resources from the named instance, 161

list-jdbc-connection-pools — gets all the JDBC connection pools from the named instance, 162
list-jmsdest — gets all the named destinations, 164
list-jndi-resources — gets all the JNDI resources from the named instance, 166
list-lifecycle-modules — gets the lifecycle modules for the named instance, 167
list-mimes — gets the MIME types for the named instance, 168
list-persistence-resources — gets all the persistence resources from the named instance, 169
list-sub-components — Lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application, 172
list-virtual-servers — gets the virtual servers in the named instance, 173
lists all the domains — list-domains, 154
lists deployed J2EE components — list-components, 151
lists one or more EJBs or Servlets in a deployed module or in a module of a deployed application — list-sub-components, 172
lists the server instances that are part of the IIOP cluster configuration — list-iiop-cluster-config, 158
lists the configurable elements — list, 146
lists the file users for the named instance — list-file-users, 156
lists the file groups for the named instance — list-file-groups, 155

M

displays the version information, 198
marks a variable name for automatic export to the environment of subsequent commands in multimode — export, 132
multimode — allows you to execute multiple commands while returning environment settings and remaining in the asadmin utility, 174

P

package-appclient — packs the application client container libraries and jar files, 175
packs the application client container libraries and jar files — package-appclient, 175
precompiles JSP source files into servlets — jspc, 144

R

reads a WAR file and the `jaxrpc-ri.xml` file and generates another WAR file that is ready for deployment — `wsdeploy`, 203
reconfig — applies the changes you have made for a server instance, 177
registers the Javamail resource to the named instance — `create-javamail-resource`, 72
registers the JDBC connection pool to the named instance — `create-jdbc-connection-pool`, 74
registers the JDBC resource to the named instance — `create-jdbc-resource`, 77
registers the JMS resource to the named instance — `create-jms-resource`, 79
registers the JNDI resource to the named instance — `create-jndi-resource`, 81
registers the named resource in the XML file specified. — `add-resources`, 14
registers the persistence resource to the named instance — `create-persistence-resource`, 88
removes one or more variables from the environment. — `unset`, 195
removes the access control list file for the named instance — `delete-acl`, 97
removes the authorized database for the named instance — `delete-authdb`, 98
removes the component from the named instance. — `undeploy`, 194
removes the custom resource from the named instance — `delete-custom-resource`, 100
removes the HTTP listener for the named instance — `delete-http-listener`, 103
removes the IIOP listener for the named instance — `delete-iiop-listener`, 107
removes the Javamail resource from the named instance — `delete-javamail-resource`, 110

removes the JDBC connection pool from the named instance — `delete-jdbc-connection-pool`, 111
removes the JDBC resource from the named instance — `delete-jdbc-resource`, 112
removes the JMS resource from the named instance — `delete-jms-resource`, 114
removes the lifecycle module for the named instance — `delete-lifecycle-module`, 118
removes the MIME type for the named instance — `delete-mime`, 119
removes the named file user — `delete-file-user`, 102
removes the persistence resource from the named instance — `delete-persistence-resource`, 120
removes the quality of service parameter for the named instance — `delete-http-qos`, 104
runs the specified component — `enable`, 131

S

set — sets the values of attributes, 181
sets the values of attributes — `set`, 181
show-component-status — displays the status of the deployed component, 182
show-instance-status — displays the status of the server instance specified., 183
shutdown — brings down the administration server and associated instances, 184
start-domain — starts the given domain, 186
starts the given domain — `start-domain`, 186
stop-domain — stops the given domain, 190
stops the given domain — `stop-domain`, 190
stops the specified component — `disable`, 129
stores the database metadata (schema) in a file for use in mapping and execution — `capture-schema`, 40

U

undeploy — removes the component from the named instance., 194
unset — removes one or more variables from the environment., 195

update-file-user — updates a current file user as specified, 196
updates a current file user as specified —
update-file-user, 196
utility for performing administrative tasks for
the Sun ONE Application Server —
asadmin, 17

V

validates the J2EE Deployment Descriptors
against application server DTDs —
verifier, 197
verifier — validates the J2EE Deployment
Descriptors against application server
DTDs, 197

W

wsdeploy — reads a WAR file and the
jaxrpc-ri.xml file and generates another
WAR file that is ready for deployment, 203