

Sun Java System Application Server Utility Reference

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

Part No: 817-6092 March 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^{TM} 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 États-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 SunTM a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

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





Contents

Preface 9

```
User Commands
                   11
add-admin-object(1)
                       12
add-resources(1)
                   14
appclient(1M)
asadmin(1M)
                18
asant(1M)
asmigrate(1m)
                 22
asupgrade(1m)
capture-schema(1m)
                       28
create-acl(1)
create-admin-object(1)
                         30
create-audit-module(1)
                         32
create-authdb(1)
create-auth-realm(1)
                       36
create-connector-connection-pool(1)
                                      38
create-connector-resource(1)
create-connector-security-map(1)
                                   43
create-custom-resource(1)
                   47
create-domain(1)
create-file-user(1)
                    49
create-http-listener(1)
                        51
create-http-qos(1)
create-iiop-listener (1)\\
                        55
create-instance(1)
                    57
```

```
create-javamail-resource(1)
create-jdbc-connection-pool(1)
                                  61
create-jdbc-resource(1)
create-jmsdest(1)
create-jmsobj(1)
create-jms-resource(1)
                         70
                         72
create-jndi-resource(1)
create-jvm-options(1)
                         74
create-lifecycle-module(1)
                             76
create-mime(1)
create-persistence-resource(1)
create-profiler(1)
create-resource-adapter-config(1)
create-ssl(1)
create-threadpool(1)
create-virtual-server(1)
                          89
delete-acl(1)
delete-admin-object(1)
                          92
delete-audit-module(1)
                          93
delete-authdb(1)
delete-auth-realm(1)
delete-connector-connection-pool(1)
                                        96
delete-connector-resource(1)
delete-connector-security-map(1)
delete-custom-resource(1)
delete-domain(1)
                     103
delete-file-user(1)
                         105
delete-http-listener(1)
delete-http-qos(1)
                     106
delete-iiop-listener(1)
                         107
delete-instance(1)
                     108
delete-javamail-resource(1)
delete-jdbc-connection-pool(1)
                                  111
delete-jdbc-resource(1)
delete-jmsdest(1)
                    114
delete-jmsobj(1)
delete-jms-resource(1)
                         116
delete-indi-resource(1)
                          117
```

⁴ Sun Java System Application Server Utility Reference • March 2004

```
delete-jvm-options(1)
delete-lifecycle-module(1)
                             120
delete-mime(1)
                  121
delete-persistence-resource(1)
delete-profiler(1)
                    123
delete-resource-adapter-config(1)
                                     124
delete-ssl(1)
               125
delete-threadpool(1)
delete-virtual-server(1)
                          128
deploy(1)
deploydir(1)
                133
deploytool(1m)
                   136
disable(1)
             137
display-license(1)
                     138
enable(1)
            139
            140
export(1)
freeze-transaction-service(1)
                               141
get(1)
       142
get-client-stubs(1)
                     144
help(1)
          145
install-license(1)
                    150
               151
jms-ping(1)
jspc(1M)
            152
        154
list(1)
list-acls(1)
             157
list-admin-objects(1)
                        158
list-audit-modules(1)
                        159
list-authdbs(1)
list-auth-realms(1)
                      161
list-components(1)
                      162
list-connector-connection-pools(1)
                                     163
list-connector-resources(1)
list-connector-security-maps(1)
                                  165
list-custom-resources(1)
list-domains(1)
                  168
list-file-groups(1)
                     169
list-file-users(1)
                   171
list-http-listeners(1)
```

```
list-iiop-listeners(1)
                       173
list-instances(1)
list-javamail-resources(1)
list-jdbc-connection-pools(1)
list-jdbc-resources(1)
                        177
list-jmsdest(1)
                 178
list-jmsobj(1)
                179
list-jms-resources(1)
                       180
list-jndi-resources(1)
                        182
list-lifecycle-modules(1)
list-mimes(1)
                184
list-persistence-resources(1)
                               185
list-profiler(1)
                 186
                  187
list-profilers(1)
list-resource-adapter-configs(1)
                                   188
list-sub-components(1)
                          190
list-threadpools(1)
list-virtual-servers(1)
                        192
multimode(1)
                           194
package-appclient(1M)
ping-connection-pool(1)
                           196
reconfig(1)
restart-instance(1)
rollback-transaction(1)
set(1)
         203
show-component-status(1)
                              204
                205
shutdown(1)
start-appserv(1)
                   206
start-domain(1)
                   207
start-instance(1)
                   208
stop-appserv(1)
                   210
stop-domain(1)
                   211
                   212
stop-instance(1)
undeploy(1)
                214
unfreeze-transaction-service(1)
                                  216
unset(1)
           217
update-connector-security-map(1)
                                      218
update-file-user(1)
```

verifier(1M) 222 verify-domain-xml(1) 223 version(1) 224 wscompile(1M) 226 wsdeploy(1M) 230

Index 233

Preface

Both novice users and those familar 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, the asadmin utility commands.
- Section 1M describes all the other Application Server utility commands.

functions.

section:

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 This section shows the syntax of commands or

The following special characters are used in this

[] Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.

Separator. Only one of the arguments separated by this character can be specified at a time.

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 secton 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 example%, or if the user must be superuser, example#. 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-admin-object(1)

NAME |

add-admin-object – adds the administered object with the specified JNDI name

SYNOPSIS

add-admin-object --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure |-s] --aorestype admin_object_type [--property name=value[:name=value]*] --raname rar_filename jndi_name

DESCRIPTION

Creates an administered object corresponding to a resource adapter. The ra.xml file can be found in the META-INF directory of the resource adapter archive. After a .RAR file is deployed to the application server, it gets extracted and a directory corresponding to the . RAR name is created under \$INSTANCE ROOT/applicaitons/j2ee-modules (or j2ee-apps). You can find the ra.xml in the META-INF subdirectory of the RAR's directory.

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).
aorestype	administered object resource type as defined by the resource adapter in the ${\tt ra.xml}$ file.
property	optional attributes name/value pairs for configuring the resource.

OPERANDS

jndi_name JNDI name of the administered object to be added.

EXAMPLES

EXAMPLE 1 Using add-admin-object

--raname

A JMS queue, called sample jmsqueue, is assumed to have already been created using the create-jmsdest command.

name of the resource adapter.

```
asadmin> add-admin-object --user admin --password adminadmin
--aorestype javax.jms.Queue --property Name=sample jmsqueue
--raname jmsra jms/samplequeue
Created the object
```

EXIT STATUS

- command executed successfully
- 1 error in executing the command

	1	- : : :	(1)
add-ad	ımın-oı	ojecti	1

 $\textbf{SEE ALSO} \ | \ \texttt{delete-admin-object(1), list-admin-objects(1)}$

44

add-resources(1)		
NAME	add-resources – registers the resource in the XML file specified	
SYNOPSIS	add-resourcesuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] xml_file_path	
DESCRIPTION	Registers the named resource in the XML file specified. The <code>xml_file_path</code> is the path to the XML file containing the resources to be registered. The DOCTYPE should be specified as <code>install_dir/lib/dtds/sun-resources_1_0.dtd</code> in the <code>resources.xml</code> file.	
	This command is sup	ported in remote mode only.
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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. Default is false.
	echo	setting to true will echo the command line statement on the standard output. Default is false.
	interactive	if set to true (default), only the required password options are prompted.
OPERANDS	<i>xml_file_path</i> path to the XML file containing the resource(s) to be registered.	
EXAMPLES	EXAMPLE 1 Using add-resources	
	asadmin> add-resourcesuser adminpasswordfile passwords.txthost localhostport 4848 resource.xml Command add-resources executed successfully	
	Where: resource.xml is the resource file containing resources to be created.	

EXIT STATUS

command executed successfully 0

1 error in executing the command

```
SEE ALSO | create-jdbc-connection-pool(1), create-jdbc-resource(1),
           create-jms-resource(1), create-jndi-resource(1),
           create-javamail-resource(1), create-persistence-resource(1),
           create-custom-resource(1)
```

appclient(1M)

NAME

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

SYNOPSIS

appclient -client client_application_jar

 $[-mainclass \ \textit{client_application_main_classname} \ | \ -name \ \textit{display_name}]$

[-xml sun-acc.xml file] [-textauth] [-user username]

[-password *password*]

DESCRIPTION

Use the appclient command to launch the application client container and invoke a client application that is packaged in an application JAR file. The application client jar file is psecified and created during deployment either by the deploytool or by using the asadmin deploy command.

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.

The client.jar that is retrieved after deploying an application, should be passed with the -client option while running the appclient utility. The -mainclass and -name options are optional for a single client application. For multiple client applications use either the -classname option or the- name option.

OPTIONS

-client	required; the name and location for the client application jar file. The
	application client JAR file is specified and created during

deployment, either by the deploytool or by the asadmin deploy

command.

-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. By default, uses the class specified in the client jar. The class name must be the full name.

For example, com.sun.test.AppClient

-name optional; the display name for the client application. Used for

multiple client applications. By default, the display name is specified in the client jar application-client.xml file which is identified

by the display-name attribute.

-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.

EXAMPLES

EXAMPLE 1 Using the appclient command

appclient -client appserv/bin/myclientapp.jar

-mainclass com.sun.test.TestAppClient -xml sun-acc.xml scott sample

EXAMPLE 1 Using the appclient command (Continued)

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

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO

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

asadmin(1M)

NAME |

asadmin – utility for performing administrative tasks for the Sun Java System Application Server

SYNOPSIS

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

DESCRIPTION

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

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

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

Remote subcommands are always executed by connecting to an administration server and executing the subcommand there. A running administration server is required. All remote subcommands require the following options:

	1 0 1
-uuser	authorized domain application server administrative username.
-wpassword	password to administer the domain application server.
-Hhost	machine name where the domain application server is running.
-pport	port number of the domain application server listening for administration requests.
-ssecure	if true, uses SSL/TLS to communicate with the domain application server.
-tterse	indicates that any output data must be very concise, typically avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false.
-eecho	setting to true will echo the command line statement on the standard output. Default is false.
-Iinteractive	if set to true (default), only the required password options are prompted.

asadmin(1M)

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 use the --secure option, you must use the set command to enable the security—enabled flag in the admin http-listener in the domain.xml.

When you use the asadmin subcommands to create and/or delete, you must restart the server for the newly created command to take affect. Use the start-domain command to restart the server.

To access the manpages for the J2EE 1.4 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.

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Unstable

SEE ALSO

appclient(1M), package-appclient(1M)

asant(1M)

NAME | asant – launches the Jakarta Ant tool

SYNOPSIS

asant target_list

DESCRIPTION

Use the asant command to automate repetitive development and deployment tasks. asant is a shell script that invokes the underlying Ant infrastructure after initializing the environment to pickup the application server installed targets.

To use Ant as part of the Sun ONE Application Server, verify that your PATH includes the provided asant (Solaris) ant.bat(Windows) script.

The bundled sample applications use asant extensively; however, asant can be used in any development or operational environments.

The build targets are represented in the build.xml files that accompany the sample applications.

To use the Ant tool to compile and reassemble the sample applications, verify that the \$AS INSTALL/bin directory is on your environment's path. On UNIX, add the \$AS INSTALL/bin directory to your PATH environment variable. On Windows, after installing the Sun ONE Application Server, set the system path by adding \$AS INSTALL\bin to the user PATH. You can access the PATH system variable from: Start menu, Settings, Control Panel, System, Advanced, Environment Variables, User Variables for Administrator, PATH.

The *target_list* is one or more space separated tasks as described below.

compiles all Java source code.

TARGETS

compile

00p0	complies an java source coue.	
jar	assembles the EJB JAR module.	
war	assembles the WAR file in <sample_dir>/assemble/war</sample_dir>	
ear	assembles the EAR file in <sample_dir>/assemble/ear</sample_dir>	
core	(default) compiles all sources, builds stubs and skeletons; and assembles EJB JAR, WAR and EAR files. This is the default target for all build.xml files shipped in the Sun ONE Application Server.	
javadocs	creates Java docs in <sample_dir>/javadocs</sample_dir>	
all	builds core and javadocs , verifies and deploys the application, and adds the resources	
deploy	deploys the application and automatically expands the EJB JAR; does not install Javadocs.	
undeploy	removes the deployed sample from the Sun ONE Application Server.	
clean	removes <appname>/build/ and <appname>/assemble/ and <appname>/javadocs directories.</appname></appname></appname>	
verify	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)
```

NAME

asmigrate – automates migration of J2EE applications from other J2EE platforms to Sun Java System Application Server

SYNOPSIS

DESCRIPTION

Use the asmigrate utility to analyze your J2EE application and translate vendor specific settings to Sun JavaTM $^{\text{TM}}$ System Application Server specific settings making the application deployable on Sun's J2EE products.

The following table identifies the supported J2EE product migrations:

Source J2EE Platform	Destination J2EE Platform
WebSphere Application Server 4.0	Sun ONE Application Server 6.5
WebLogic Application Server 5.1	
WebLogic Application Server 5.1, 6.0, 6.1	Sun ONE Application Server 7
WebSphere Application Server 4.0	
JavaTM TM 2 Platform Enterprise Edition 1.3	
Sun ONE Application Server 6.x	
Sun ONE Web Server 6.0	
JBoss Application Server 3.0	
Tomcat Web Server 4.1.12	

Source J2EE Platform	Destination J2EE Platform
WebLogic Application Server 5.1, 6.0, 6.1	JavaTM™ 2 Platform, Enterprise Edition 1.4
WebSphere Application Server 4.0	Application Server
JavaTM [™] 2 Platform Enterprise Edition 1.3/1.4	
Sun ONE Application Server 6.x	
Sun ONE Web Server 6.0	
JBoss Application Server 3.0	
Tomcat Web Server 4.1.12	
WebLogic Application Server 5.1, 6.0, 6.1	Sun JavaTM TM System Application Server Platform
WebSphere Application Server 4.0	Edition 8
JavTM™a 2 Platform Enterprise Edition 1.3/1.4	
Sun ONE Application Server 6.x	
Sun ONE Web Server 6.0	
JBoss Application Server 3.0	
JBoss Application Server 3.2	
Tomcat Web Server 4.1.12	

OPTIONS

-hhelp	displays the arguments for launching the MigrationTool.
-vversion	displays the version of the MigrationTool.
-uui	invokes the tool in user interface mode.
-ccommandline	invokes the tool in command-line mode.
-qquiet	launches the tool in quiet mode.
-ddebug	launches the tool in debug mode.
-ssourcedirectory	identifes the directory where the source code to migrate or scan is present.
-Ssourceserver	identifes the source application server of the applications to be migrated. Possible servers include:
	 wl51: WebLogic Application Server 5.1 wl60: WebLogic Application Server 6.0 wl61: WebLogic Application Server 6.1 as65: Sun ONE Application Server 6.5 as70: Sun ONE Application Server 7.0

	 ws40: WebSphere Application Server 4.0 ri13: JavaTMTM 2 Platform Enterprise Edition 1.3 ri14: JavaTMTM 2 Platform Enterprise Edition 1.3 s1ws: Sun ONE Web Server jb30: JBoss Application Server 3.0 tc41: Tomcat Application Server 4.1
-ttargetdirectory	target or output directory where the migrated application should be placed.
-Ttargetserver	target application server to which the application is to be migrated.
-nscan-native-apis-only	scans the source code only for the presence of application server specific proprietary APIs.
-pscan-packages	comma-separated list of Java packages to scan.
-jjava2db	bypasses the creation of the sun-cmp-mapping.xml file. Instead, introduces the option argument into the sun-ejb-jar.xml file. Option arguments are:
	 create-tables: if set to true (default), creates tables at deploy. If set to false tables are not created. drop-tables: if set to true (default), tables are dropped at undeploy. If set to false tables are not dropped. db-vendor-name: name of the database vendor for the application to be migrated. Supported vendor names include: Oracle, Sybase, DB2, Generic SQL92, PointBase, MSSQL.
-mmigrate-cmp	migrates 1.1 compliant CMPs, if any, to 2.0. Option arguments are:
-ffile-filter	 overwrite-conflicting-accessors: if set to true (default), conflicting accessors are overwritten. If set to false, conflicting accessors are not overwritten. comment-pk-modifiers: if set to true (default), setters of primary key are commented. If set to false, setters of primary key are not commented. selects the type of files to migrate. Option
	arguments are:

- all-files: if specified and set to true (default), migrates all types of files.
 - html-files: if specified and set to true (default), migrates HTML files.
- java-files: if specified and set to true (default), migrates Java files.
- jsp-files: if specified and set to true (default), migrates JSP type files.
- archive-files: if specified and set to true (default), migrates jar/ear/war/rar file types.

if specified, appends the logging to the existing or previous logs without overwriting them. If not specified, previous logs are overwritten.

identifes the archive file (jar/ear/war/rar) to be migrated.

-a --append-logs

operands

SEE ALSO asupgrade(1M)

asupgrade(1m)

NAME

asupgrade – migrates the configuration of a previously installed Sun Java System Application Server

SYNOPSIS

```
asupgrade [-c | --console ] [-V --version ] [-h | --help ]-s |
    -source applicationserver7.x_installation
    -t --target applicationserver8.x_installation [-d | --domain domain_name
    -n | --nsspwdfile NSS_password_filepath -j |
    --jkspwdfile JKS_password_filepath -p |
    --capwdfile CA_password_filepath]
```

DESCRIPTION

Use the asupgrade utility to migrate the server configuration and its persisted state, J2EE services, and deployed J2EE applications. The configuration of an installed Sun Java System Application Server 7 is migrated to the Sun Java System Application Server 8 Application Server installation. If the domain contains information about a deployed application and the installed application components do not agree with the configuration information, the configuration is migrated as is without any attempt to reconfigure the incorrect configurations.

asupgrade migrates the configuration and deployed applications of a previous version of the Application Server; however, the runtime binaries of the server are not updated. Database migrations or conversions are beyond the scope of the asupgrade command.

Only those instances that do not use the Sun Java System Web Server specific features will be upgraded seamlessly. Configration files related to HTTP path, CGI bin, SHTML, and NSAPI plugins will not be upgraded.

The upgrade process can also be initiated automatically at installation time using the Upgrade checkbox in the Application Server installer. After completion of the upgrade, use the Application Server 7 Uninstaller to remove the previous version of the Application Server.

Application archives (.ear) and component archives (.jar, .war, .rar) that are deployed in the Application Server 7 environment do not require any modification to run on Application Server 8. However applications and components deployed in the source server are repackaged into new J2EE archives in the target server's autodeploy directory and are deployed upon server startup. Applications that do not deploy successfully, must use Migrationtool (asmigrate) on the application and then manually redeploy the application.

You must specify the source and target directories for the upgrade. If the upgrade includes certificates, you must also provide the passwords for the source PKCS12 file and the target JKS keyfile for each domain that contains certificates to be migrated. Since Application Server 7 uses a different certificate store format (NSS) than Application Server 8 (JSSE), the migration keys and certificates are converted to the new format. Upon successful upgrade, an upgrate report is generated listing successfully migrated items along with a list of the items that could not be migrated.

OPTIONS

-c --console launches the upgrade command line utility.

asupgrade(1m)

-Vversion	displays the version of the UpgradeTool.
-hhelp	displays the arguments for launching the UpgradeTool.
-ssource	identifies the installation directory for Sun Java System Application Server 7.
-ttarget	identifies the installation directory for Sun Java System Application Server 8.
-ddomain	identifies the destination domain name for the migrated certificates.
-nnsspwdfile	identifies the path to the NSS password file.
-jjkspwdfile	identifies the path to the JKS password file.
-pcapwdfile	identifies the path to the CA certificate password file.

EXAMPLES

EXAMPLE 1 Using asupgrade

example% upgrade -s /home/sunas7 -t /home/sunas8

SEE ALSO

asmigrate(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 url -username name- password password-driver
 a_jdbc_driver [-schemaname name] [-table tablename] * [-out filename]

DESCRIPTION

Stores the database metadata (schema) in a file. You can also use the Sun Java System Studio IDE to capture the database schema. Run capture-schema as the same database user that owns the tables, so that the -username and -schemaname options are given the same username. Additionally, you can grant the database user running the capture-schema command the ANALYZE ANY TABLE privilege.

OPTIONS

-dburl JDBC URL expected by the driver for accessing a database.

-username user name for authenticating access to a database.

-password password for accessing the selected database.-driverJDBC driver classname in your CLASSPATH.

-schemaname name of the user schema being captured. If not specified, the

default will capture metadata for all tables from all the schemas accessible to this user. Specifying this parameter is highly recommended. If more than one schema is accessible to this user, more than one table with the same name may be captured which

will cause problems.

-table name of the table; multiple table names can be specified.

-out output target; defaults to stdout. This parameter corresponds to

the schema sub-element of the sun-cmp-mapping element in the

sun-cmp-mapping 1 1.dtd file.

EXAMPLES

EXAMPLE 1 Using capture-schema

capture-schema -dburl jdbc:oracle:thin:@sadbuttrue:1521:ora817
-schemaname cantiflas -username CANTIFLAS -password enigma -driver
oracle.jdbc.driver.OracleDriver

SEE ALSO

asadmin(1M)

create-acl(1)

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

SYNOPSIS

create-acl

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

DESCRIPTION

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

OPTIONS

--user administrative user associated for the instance.

administrative password corresponding to the administrative --password

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

administrative port number associated with the administrative --port

--secure indicates communication with the administrative instance in

secured mode.

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

administrative instance).

name of the instance. --instance

name of the default acl file. --aclfile

OPERANDS

internal name for the ACL file listing. This ID is used in a virtual server acl ID

element to define the ACL file used by the virtual server.

EXAMPLES

EXAMPLE 1 Using create-acl

asadmin> create-acl --user admin --password adminadmin --host fuyako --port 7070 --instance serve Created ACL with id=sampleACL

Where: sampleACL is the name of the ACL created.

EXIT STATUS

- 0 command executed successfully
- error in executing the command

INTERFACE EQUIVALENT SEE ALSO Access Control List page

delete-acl(1), list-acl(1)

create-admin-object(1)

NAME |

create-admin-object - adds the administered object with the specified JNDI name

SYNOPSIS

create-admin-object --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --restype aorestype --raname resource_adapter_name [--enabled=true] [--description description] [--property name=value[:name=value]*] jndi_name

DESCRIPTION

Creates an administered object specific to the Enterprise Information Server (typically a messaging provider) with the JNDI name provided. The resource adapter exposes the administered object as a Javabean. Application programs can lookup the administered object using the JNDI name, and use it using messaging style specific APIs.

Before you can create an administered object, you must first deploy the resource adapter and specify it using the --raname option. This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.	
password	password to administer the domain application server.	
host	machine name where the domain application server is running.	
port	port number of the domain application server listening for administration requests.	
secure	if true, uses SSL/TLS to communicate with the domain application server.	
passwordfile	file containing the domain application server password.	
terse	indicates that any output data must be very concise, typical avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false.	
echo	setting to true will echo the command line statement on the standard output. Default is false.	
interactive	if set to true (default), only the required password options are prompted.	
restype	administered object resource type as defined by the resource adapter in the ra.xml file. This is the same value found in the adminobject-interface elements in the deployment descriptor of a resource adapter (ra.xml).	

create-admin-object(1)

module name of the deployed resource adapter. In case of a --raname

resource adapter that is embedded in an application, the

raname will be appname#raname.

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

text description of the connection pool. --description

name/value pair of the properties of admin object java --property

> bean. admin object java bean is specified by the adminobject-class in the resource adapter's deployment

descriptor (ra.xml).

OPERANDS jndi_name

JNDI name of the administered object to be created.

EXAMPLES EXAMPLE 1 Using create-admin-object

> The javax.jms.Queue resource type is obtained from the rar.xml file. The jmsra must be deployed prior to executing this command.

asadmin> create-admin-object --user admin1 --password adminadmin1 --restype javax.jms.Queue --raname jmsra --property Name=sample_jmsqueue jms/samplequeue Command create-admin-object executed successfully

EXIT STATUS

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

SEE ALSO

delete-admin-object(1), list-admin-objects(1)

create-audit-module(1)

NAME	E create-audit-module – adds an audit-module		
SYNOPSIS	<pre>create-audit-moduleuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]classname realm_class [property (name=value) [:name=value]*] audit_module_name</pre>		
DESCRIPTION	Adds the named audit module for the plugin module that implements the audit capabilities. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	classname	Java class which implements this realm.	
	property	optional attributes name/value pairs of provider implementation specific attributes.	
OPERANDS	audit_module_name	name of this audit module.	
EXAMPLES	EXAMPLE 1 Using create-audit-module asadmin> create-audit-moduleuser admin1password adminadmin1host pigeonport 5001classname com.sun.appserv.auditmoduleproperty defaultuser=admin:Password=admin sampleAuditModule Command create-audit-module executed successfully		
EXIT STATUS	0 command executed successfully		
	1 error in executin	•	

create-audit-module(1)

 $\textbf{SEE ALSO} \ | \ \texttt{delete-audit-module}(1), \texttt{list-audit-modules}(1)$

create-authdb(1)

NAME	NAME create-authdb - adds the new authorized database for the named instance SYNOPSIS create-authdb cre			
SYNOPSIS				
DESCRIPTION	Adds the named authorized database associated with the named server instance.			
OPTIONS	user	administrative user associated for the instance.		
	password	administrative password corresponding to the administrative user.		
	host	host name of the machine hosting the administrative instance.		
	port	administrative port number associated with the administrative host.		
	passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).		
	secure	if true, uses SSL/TLS to communicate with the administrative instance.		
	instance	name of the instance.		
	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.		
OPERANDS <i>authdb_id</i> user database name in the virtual server's ACL file.		base name in the virtual server's ACL file.		
EXAMPLES	EXAMPLE 1 Using create-authdb			
	asadmin> create-authdbuser adminpassword adminadminhost fuyakoport 7070 Created AuthDB with id = sampleAuth			
	Where sampleAuth is the authdb created.			
EXIT STATUS	0 command executed successfully			

create-authdb(1)

error in executing the command 1

INTERFACE EQUIVALENT SEE ALSO

unknown

delete-authdb(1), list-authdbs(1)

create-auth-realm(1)

NAME	create-auth-realm – adds the new authorized realm		
SYNOPSIS	<pre>create-auth-realmuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]classname realm_class [property (name=value)[:name=value]*] auth_realm_name</pre>		
DESCRIPTION	Adds the named auth	orized realm. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	classname	Java class which implements this realm.	
	property	optional attributes name/value paris of provider implementation specific attributes.	
OPERANDS	auth_realm_name r	name of this realm.	
EXAMPLES	EXAMPLE 1 Using create-auth-realm		
	asadmin> create-auth-realmuser admin1password adminadmin1host pigeonport 5001classname com.iplanet.ias.security.auth.realm.DB.Databaproperty defaultuser=admin:Password=admin db Command create-auth-realm executed successfully		
	Where db is the auth realm created.		
EXIT STATUS	6 0 command executed successfully		
1 error in executing the command			

create-auth-realm(1)

 $\textbf{SEE ALSO} \ | \ \texttt{delete-auth-realm}(1), \texttt{list-auth-realms}(1)$

create-connector-connection-pool(1)

NAME	create-connector-connection-pool	l – adds a connection pool with the specified	
SYNOPSIS	create-connector-connection-pooluser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] [steadypoolsize 8] [maxpoolsize 32] [maxwaittime 3000] [poolresize 2] [idletimeout 300] [failallconnections=false]raname resource_adapter_nameconnectiondefinition connection-definition_name [property name=value[:name=value]*] pool_name		
DESCRIPTION		adds a new connector connection pool. Before you can add a new connector connection pool, you must first deploy the associated RAR file.	
	This command is supported in re	emote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	steadypoolsize	minimum and initial number of connections maintained in the pool (default is 8).	
	maxpoolsize	maximum number of connections that can be created to satisfy client requests (default is 32).	
	maxwaittime	amount of time, in milliseconds, that a caller will have to wait before a connection is created if a	

create-connector-connection-pool(1)

	connection is not available. If set to 0 the caller is blocked indefinately until a resource is available or an error occurs (default is 60000).
poolresize	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 the steadypoolsize, the connection removal stops.
idletimeout	the maximum time that a connection can remain idle in the pool. After this amount of time, the pool can close this connection (default is 300).
failallconnections	if true, closes all connections in the pool if a single validation check fails. This parameter is mandatory only if the is-connection-validation-required is set to true. Legal values are: on, off, yes, no, 1, 0, true or false (default is false).
raname	module name of the deployed resource adapter. In case of a resource adapter that is embedded in an application, the raname will be appname#raname.
connectiondefinition	identifies one of the connection definitions in the deployment descriptor of a resource adapter. Value of connectionfactory-interface sub-element in the connection-definition element.
property	name/value pair of the configuration properties of the managedconnectionfactory-class of the connectiondefinition for which the pool is being created. The UserName and Password are accepted by default for all pools. The escape character "\" is used in thisproperty option to distinguish the colons (:) and the backslash(/).

OPERANDS

pool_name name of the connection pool to be created.

EXAMPLES

EXAMPLE 1 Using create-connector-connection-pool

The JMS resource adapter that is prepackaged with the application server is used for this example. Before you can add a new connector connection pool, you must first deploy the associated RAR file.

To deploy the RAR file:

```
asadmin> deploy --user admin --password adminadmin jmsra.rar
Command deploy executed successfully
asadmin> create-connector-connection-pool --user admin1
--password adminadmin1 --steadypoolsize 20 --maxpoolsize 100 --poolresize 2
```

create-connector-connection-pool(1)

EXAMPLE 1 Using create-connector-connection-pool (Continued)

--maxwait 60000 --raname jmsra --connectiondefinition javax.jms.QueueConnectionFactory
--property UserName=guest:Password=guest jms/qConnPool

Command create-connector-connection-pool executed successfully

EXIT STATUS

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

SEE ALSO

$$\label{eq:connection-pool} \begin{split} \text{deploy}(1), \, \text{delete-connector-connection-pool}(1), \\ \text{list-connector-connection-pools}(1) \end{split}$$

		create connector resource(1)	
NAME	create-connector-resource – registers the resource with the specified JNDI name		
SYNOPSIS	<pre>create-connector-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] poolname connector_connection_pool_name [enabled=true] [description description] jndi_name</pre>		
DESCRIPTION	Registers the resource with the specified JNDI name. The connector connection pool must already exist. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	poolname	name of the connection pool connector resource. If two or more resource elements point to the same connection pool element, they will use the same pool connections at runtime.	
	enabled	determines whether the resource is enabled at runtime. If a resource is not enabled, you are not able to use the resource. Use the set command to enable or disable resources.	
	description	text description of the connection pool.	
OPER ANDC	'1' INIDI		

OPERANDS

JNDI name of the resource to be created. jndi_name

EXAMPLES

EXAMPLE 1 Using the create-connector-resource command

 $\label{prop:connection-pool} Execute the example described for \verb|create-connection-pool|| \\$ command before trying out this example.

create-connector-resource(1)

EXAMPLE 1 Using the create-connector-resource command (*Continued*)

asadmin> create-connector-resource --user admin1 --password adminadmin1 --host pigeon --port 5001 --poolname jms/qConnPool --description "creating a sample connector resource" jms/qConnFactory

Command create-connector-resource executed successfully

EXIT STATUS

- command executed successfully
- 1 error in executing the command

SEE ALSO

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

NAME |

create-connector-security-map – creates a security map for the named connector connection pool

SYNOPSIS

```
create-connector-security-map --user admin_user
```

[--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --poolname

connector_connection_pool_name

--principals principal-name[, principal-name]*| --usergroups user-group[, user-group]* [--mappedpassword password] mapname

authorized domain application server administrative

DESCRIPTION

Creates a security map for the named connector connection pool. If the security map is not present, one is created. You must have first created a connector connection pool using the create-connector-connection-pool command. The enterprise information system is any system which holds the information. It can be a mainframe, a messaging system, a database system, or even an application.

The --principals option and --usergroups option are mutually exclusive; only one should be used.

This command is supported in remote mode only.

OPTIONS

--user

abel	username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
poolname	connector connection pool name.
principals	a comma separated list of J2EE principals.
usergroups	a comma separated list of J2EE usergroups.

create-connector-security-map(1)

--mappedusername the enterprise information system username.

--mappedpassword the enterprise information system password.

OPERANDS

mapname name of the security map to be created.

EXAMPLES

EXAMPLE 1 Using create-connector-security-map

It is assumed that the connector pool has already been created using the create-connector-pool command.

asadmin> create-connector-security-map --user admin --password adminadmin poolname connector-pool1 --principals principal1,principal2

--usergroups usergroup1,usergroup2 --mappedusername backend-username

--mappedpassword backend-password securityMap1

Command create-connector-security-map executed successfully

EXIT STATUS

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

SEE ALSO

delete-connector-security-map(1), list-connector-security-map(1), update-connector-security-map(1)

```
NAME |
                 create-custom-resource – registers the custom resource
    SYNOPSIS
                 create-custom-resource --user admin_user [--password admin_password]
                       [--host localhost] [--port 4848] [--secure | -s]
                       [--passwordfile filename ] [--terse=false] [--echo=false]
                       [--interactive=true] --restype resource_type
                       --factoryclass classname [--enabled=true] [--description text]
                       [--property (name=value)[:name=value]*] jndi_name
DESCRIPTION
                 Registers the custom resource. This command is supported in remote mode only.
    OPTIONS
                                       authorized domain application server administrative
                 --user
                                       username.
                 --password
                                       password to administer the domain application server.
                 --host
                                       machine name where the domain application server is running.
                                       port number of the domain application server listening for
                 --port
                                       administration requests.
                 --secure
                                       if true, uses SSL/TLS to communicate with the domain
                                       application server.
                                       file containing the domain application server password.
                  --passwordfile
                                       indicates that any output data must be very concise, typically
                 --terse
                                       avoiding human-friendly sentences and favoring
                                       well-formatted data for consumption by a script. Default is
                                       false.
                 --echo
                                       setting to true will echo the command line statement on the
                                       standard output. Default is false.
                 --interactive
                                       if set to true (default), only the required password options are
                                       prompted.
                  --restype
                                       type of custom resource to be created.
                 --factoryclass
                                       class that creates the custom resource.
                 --enabled
                                       determines whether the resource is enabled at runtime.
                                       text description of the custom resource.
                  --description
                                       optional attribute name/value pairs for configuring the custom
                 --property
                                       resource.
  OPERANDS
                 jndi_name
                              INDI name of the custom resource to be created.
   EXAMPLES
                 EXAMPLE 1 Using create-custom-resources
                 asadmin> create-custom-resource --user admin1 --password adminadmin1
                  --host pigeon --port 5001 --restype customType factoryclass "com.custom.class" --description
                 "this is a sample of creating a custom resource" sample custom resource
                 Command create-custom-resource executed successfully
```

create-custom-resource(1)

EXAMPLE 1 Using create-custom-resources (Continued)

Where ${\tt sample_custom_resource}$ is the custom resource created.

EXIT STATUS

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

SEE ALSO

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

NAME | create-domain – creates a domain with the given name

SYNOPSIS

create-domain [--domaindir install_dir/domains] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=false] --adminport port_number --adminuser admin_user [--adminpassword password] [--instanceport 8080] [--domainproperties (name=value) [:name=value]*] domain_name

DESCRIPTION

create-domain creates 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 is supported in local mode only.

If a user creates a domain in a non-default directory, the domain will not be automatically shutdown during uninstallation.

OPTIONS

•	8
domaindir	directory where the domain is to be created. If specified, path must be accessible in the filesystem. If not specified, the domain is created in the default <code>install_dir/domains</code> directory.
passwordfile	file containing the domain application server password in the following form: AS_ADMIN_PASSWORD= <i>password</i> . Where <i>password</i> is the actual administrator password for the domain.
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. Default is false.
echo	setting to true will echo the command line statement on to the standard output. Default is false.
interactive	if set to true (default), only the required options are prompted.
adminport	port number of the domain application server listening for the administration requests.
adminuser	administrative username for the specified domain.
adminpassword	password to administer the domain application server.
instanceport	port number of the http listener. The port number cannot be currently in use.
domainproperties	list of the property name/value pairs for the domain to be created separated by the ":" character. The property value assignment character is =. Valid property names are: jms.port, orb.listener.port, http.ssl.port, orb.ssl.port, orb.mutualauth.port. Valid values

create-domain(1)

```
http.ssl.port=1043:orb.ssl.port=1060:orb.mutualauth.port=
 OPERANDS
               domain_name
                              name of the domain. Must be a unique name.
  EXAMPLES
               EXAMPLE 1 Using create-domain
               asadmin> create-domain --domaindir /export/domains
               --adminport 7070 --adminuser admin --adminpassword adminadmin
               --instanceport 7071 sampleDomain
               created domain sampleDomain successfully
               Where: the sampleDomain domain is created in the /export/domains directory.
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)
```

NAME | create-file-user – creates a new file user

SYNOPSIS

create-file-user --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--userpassword user_password] [--groups user_groups:[user_groups]*] [--authrealmname auth_realm_name] user_name

DESCRIPTION

Creates an entry in keyfile with the specified username, userpassword, and groups. Multiple groups can be created by separating them with a colon ":". If the auth_realm_name is not specified, an entry is created in the default keyfile. If auth_realm_name is specified, an entry is created in the keyfile where the auth-realm name in the domain.xml file points to.

authorized domain application server administrative

This command is supported in remote mode only.

OPTIONS

--user

	username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
userpassword	password for the file user.
groups	group where the file user belongs to.
authrealmname	name, in the domain.xml file, where you have different stores for file auth realm.

OPERANDS

name of file user to be created. user_name

create-file-user(1)

EXAMPLES

EXAMPLE 1 Using the create-file-user command

It is assumed that an authority realm has already been created using the create-auth-realm command.

```
asadmin> create-file-user --user admin1 --password adminadmin1 --host pigeon --port 5001 --userpassword sample --groups staff:manager --authrealmname auth-realm1 sample_user

Command create-file-user executed successfully
```

Where: the sample_user is the file user created.

EXIT STATUS

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

SEE ALSO

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

NAME	create-http-listener – adds a new HTTP listener socket		
SYNOPSIS	create-http-listeneruser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename]listeneraddress address [terse=false] [echo=false] [interactive=true]listenerport listener_portdefaultvs virtual_serverservername server_name [acceptorthreads acceptor_threads] [securityenabled=false] [enabled=true] listener_ID		
DESCRIPTION	Creates the HTTP listener. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	listeneraddress	IP address of the listener address.	
	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	

create-http-listener(1)

port number is appended, that port will be used in URLs

that the server sends to the client.

--acceptorthreads number of acceptor threads for the listen socket. The

recommended value is the number of processors in the

machine.

--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 globally enables or disables SSL by making certificates available to the server instance.

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

OPERANDS

listener_id listener ID of the HTTP listener.

EXAMPLES

EXAMPLE 1 Using create-http-listener

asadmin> create-http-listener --user admin1 --password adminadmin1

--host pigeon --port 5001 --listeneraddress 0.0.0.0

--listenerport 7272 --defaultvs server --servername pigeon.red.iplanet.com

 $\hbox{--acceptorthreads 2 --security enabled=false --enabled=false sample Listener} \\$

Command create-http-listener executed successfully

Where: sampleListener is the HTTP listener created.

EXIT STATUS

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

SEE ALSO

delete-http-listener(1), list-http-listeners(1)

NAME | create-http-qos – creates a new quality of service parameter

SYNOPSIS

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

DESCRIPTION

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

OPTIONS

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.

file containing passwords appropriate for the command --passwordfile

(e.g., administrative instance).

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

administrative instance.

virtual server ID. It can also be referred to as the variable --virtualserver

\$id in an obj.conf file. A virtual server ID cannot begin

with a number.

--bwlimit maximum bandwidth limit, for the virtual server class or

virtual server, in bytes per second. The default is no limit.

determines whether the bandwidth limit should be --enforcebwlimit

enforced or not.

maximum number of concurrent connections for the --connlimit

server, virtual server class, or virtual server.

determines whether the connection limit should be --enforceconnlimit

enforced or not.

OPERANDS

instance_name name of the instance.

EXAMPLES

EXAMPLE 1 Using create-http-qos

asadmin> create-http-qos --user admin --password adminadmin --host fuyako --port 7070 --bwlimit 10 Created HTTP QOS

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

create-http-qos(1)

EXIT STATUS | 0

- command executed successfully
- 1 error in executing the command

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

SEE ALSO

delete-http-qos(1)

```
SYNOPSIS
                 create-iiop-listener --user admin_user [--password admin_password]
                       [--host localhost] [--port 4848] [--secure | -s]
                       [--passwordfile filename] [--terse=false] [--echo=false]
                       [--interactive=true] --listeneraddress address [--iiopport
                       iiop_port] [--enabled=true] [--property (name=value)[:name=value]*]
                       listener_ID
DESCRIPTION
                 Adds the IIOP listener. This command is supported in remote mode only.
    OPTIONS
                                          authorized domain application server administrative
                 --user
                                          username.
                 --password
                                          password to administer the domain application server.
                 --host
                                          machine name where the domain application server is
                                          running.
                 --port
                                          port number of the domain application server listening for
                                          administration requests.
                 --secure
                                          if true, uses SSL/TLS to communicate with the domain
                                          application server.
                 --passwordfile
                                          file containing the domain application server password.
                                          indicates that any output data must be very concise,
                 --terse
                                          typically avoiding human-friendly sentences and favoring
                                          well-formatted data for consumption by a script. Default is
                                          false.
                 --echo
                                          setting to true will echo the command line statement on the
                                          standard output. Default is false.
                 --interactive
                                          if set to true (default), only the required password options
                                          are prompted.
                                          can be the IP address or the hostname
                 --listeneraddress
                 --iiopport
                                          IIOP port number.
                 --enabled
                                          determines whether the IIOP listener is enabled at runtime.
                                          optional attribute name/value pairs for configuring the
                 --property
                                          resource.
  OPERANDS
                 listener_id
                              unique identifier for the IIOP listener to be created.
   EXAMPLES
                 EXAMPLE 1 Using create-iiop-listener
                 asadmin> create-iiop-listener --user admin --password adminadmin
                 --host fuyako --port 7070 --listeneraddress 192.168.1.100 --iiopport 8080
                 sample iiop listener
                 Created IIOP listener with id = sample iiop listener
```

create-iiop-listener – adds the IIOP listener

NAME

create-iiop-listener(1)

EXAMPLE 1 Using create-iiop-listener (Continued)

Where: $sample_iiop_listener$ is the IIOP listener created.

EXIT STATUS

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

SEE ALSO

delete-iiop-listener(1), list-iiop-listeners(1)

NAME |

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

SYNOPSIS

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

DESCRIPTION

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

OPTIONS

an definition of server is arready running for the specifica hostitative, then the
system defaults to the local hostname. To create the instance locally, not requiring the
administration server to be up and running, specify thelocal option. The named
instance must not exist within that domain.

administrative user associated for the instance. --user

administrative password corresponding to the administrative --password

user.

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

--port administrative port number associated with the administrative

host.

owner of the domain directory. --sysuser

--domain name of the domain.

--local determines if the command should delegate the request to

administrative instance or run locally.

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

administrative instance).

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

instance.

--instanceport port where the instance listens for requests.

OPERANDS

name of the instance to be created. instance_name

EXAMPLES

EXAMPLE 1 Using create-instance in local mode

asadmin> create-instance --domain domain1 --instanceport 8967 --sysuser adminuser server4 Created Instance server4 successfully

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

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
```

create-instance(1)

EXAMPLE 2 Using create-instance in remote mode (Continued)

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

INTERFACE EQUIVALENT SEE ALSO

Application Server instances page

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

NAME create-javamail-resource – registers the JavaMail resource **SYNOPSIS** create-javamail-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive] --mailhost hostname --mailuser username --fromaddress address [--storeprotocol imap] [--storeprotocolclass com.sun.mail.imapIMAPStore] [--transprotocol smtp] [--transprotocolclass com.sun.mail.smtp.SMTPTransport] [--debug=false][--enabled=true] [--description text] [--property (name=value)[:name=value]*] jndi_name **DESCRIPTION** Registers the Javamail resource. This command can only be run remotely. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening --port for administration requests. if true, uses SSL/TLS to communicate with the domain --secure application server. file containing the domain application server --passwordfile password. --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. setting to true will echo the command line statement --echo on the standard output. --interactive prompts you for the required options that are not already specified. --mailhost mail server host. --mailuser mail account user name. --fromaddress email address. --storeprotocol mail server stored protocol. mail server stored protocol class name. --storeprotocolclass --transprotocol mail server transport protocol.

create-javamail-resource(1)

--transprotocolclass mail server transport protocol class name.

--debug if set to true, server startsup in debug mode for this

resource.

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

--description text description of the Javamail resource.

--property optional attribute name/value pairs for configuring the

Javamail resource.

OPERANDS

jndi_name JNDI name of the Javamail resource to be created.

EXAMPLES

EXAMPLE 1 Using create-javamail-resource

asadmin> create-javamail-resource --user admin

--password adminadmin --host fuyako --port 7070 --mailhost localhost

 $--mailuser \ sample \ --from address \ sample \verb|\@sun|.com \ mail_resourc/MyMailSession$

Command create-javamail-resource executed successfully

Where: mail/MyMailSession is the JavaMail resource created. The escape character (\) is used in the --fromaddress option to distinguish the dot (.) and at sign (@). The JNDI name for a JavaMail resource customarily includes the mail/ naming subcontext.

EXIT STATUS

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

SEE ALSO

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

NAME

create-jdbc-connection-pool – registers the JDBC connection pool

SYNOPSIS

create-jdbc-connection-pool --user admin_user

[--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false]

[--echo=false] [--interactive=true]

--datasourceclassname classname [--restype res_type]

[--steadypoolsize 8] [--maxpoolsize 32] [--maxwait 6000]

[--poolresize 2] [--idletimeout 300]

[--isolationlevel isolation_level] [--isisolationguaranteed]

[--isconnectvalidatereq=false]

[--validationmethod auto-commit] [--validationtable table_name]

[--failconnection=false] [--description text]

[--property (name=value)[:name=value]*] connection_pool_ID

DESCRIPTION

Registers the JDBC connection pool. All the properties of the connection pool can be modified dynamically at runtime. If you change any of the pool properties (for example the maxpoolsize), the change is effected without any change to the existing connections. If you change any of the connection related properties (such as UserName), then all the existing connections in the pool are destroyed and new connections will be created. Use the set command to change any property.

authorized domain application server

This command is supported in remote mode only.

OPTIONS

--user

	administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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.

create-jdbc-connection-pool(1)

1 ' '	
datasourceclassname	name of the vendor supplied JDBC datasource resource manager.
restype	must be specified to disambiguate when a datasource class implements more than one of the JDBC interfaces <code>javax.sql.DataSource</code> , <code>javax.sql.ConnectionPoolDataSource</code> or <code>javax.sql.XADataSource</code> . An error is produced when this option has a legal value and the indicated interface is not implemented by the datasource class.
steadypoolsize	minimum and initial number of connections maintained in the pool.
maxpoolsize	maximum number of connections that can be created.
maxwait	the amount of time a caller will wait before a connection timeout is sent. The default is 60 seconds. A value of 0 forces the caller to wait indefinitely.
poolresize	number of connections to be removed when idletimeout timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches steadypoolsize, the connection removal stops.
idletimeout	maximum time (in seconds) that a connection can remain idle in the pool. After this time, the implementation can close this connection. It is recommended that this timeout is kept shorter than the server side timeout to prevent the accumulation of unusable connections in the application.
isolationlevel	specifies the transaction-isolation-level on the pooled database connections. This option does not have a default value. If not specified, the pool operates with default isolation level provided by the JDBC driver. A desired isolation level can be set using one of the standard transaction isolation levels: read-uncommitted, read-committed, repeatable-read, serializable. Applications that change the isolation level on a pooled connection programmatically risk polluting the pool. This could lead to program errors.
isisolationguaranteed	applicable only when a particular isolation level is specified for transaction-isolation-level. The default

create-jdbc-connection-pool(1)

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. Can be set to false by the administrator when confident that the application does not change the isolation level before returning

the connection.

if set to true connections are validated (checked to --isconnectvalidatereg

see if they are usable) before giving out the

application. The default is false.

--validationmethod name of the validation table used to perform a

> query to validate a connection. Valid settings are: auto-commit, meta-data, or table. Defaults to

auto-commit.

--validationtable name of the validation table used to perform a

> query to validate a connection. This parameter is mandatory if connection-validation-type is set to table. Verification by accessing a user specified table may become necessary for connection

validation.

--failconnection if set to true, all connection in the pool must be

> closed if a single validation check fails; defaults to false. One attempt is made to re-establish failed

connections.

--description text description of the JDBC connection pool.

optional attributes name/value pairs for --property

configuring the connection pool.

OPERANDS

connection_pool_id name of the JDBC connection pool to be created.

EXAMPLES

EXAMPLE 1 Using create-jdbc-connection-pool

asadmin> create-jdbc-connection-pool --user admin --password adminadmin

- --host fuyako --port 7070 --datasourceclassname com.poiintbase.jdbc.jdbcUniversalDriver --restype jax.sql.XADataSource --isolationlevel serializable --isconnectvalidatereq=true
- --validationmethod auto-commit --description "XA Connection"
- --property DatabaseName="jdbc\:pointbase\:server\:\/\/localhost:9093\/sample"

:User=public:Password=public XA connection pool

Command create-jdbc-connection-pool executed successfully

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

EXIT STATUS

command executed successfully 0

create-i	idhc-	connection-pool(′1 `
CICatc-	ubc-		1

1 error in executing the command **SEE ALSO** delete-jdbc-connection-pool(1), list-jdbc-connection-pools(1)

```
SYNOPSIS
                 create-jdbc-resource --user admin_user [--password admin_password]
                       [--host localhost] [--port 4848] [--secure | -s]
                       [--passwordfile filename] [--terse=false] [--echo=false]
                       [--interactive=true] --connectionpoolid ID [--enabled=true]
                       [--description text] jndi_name
DESCRIPTION
                 Registers the JDBC resource. This command is supported in remote mode only.
    OPTIONS
                 --user
                                           authorized domain application server administrative
                                            username.
                 --password
                                           password to administer the domain application server.
                 --host
                                            machine name where the domain application server is
                                           running.
                 --port
                                           port number of the domain application server listening
                                           for administration requests.
                 --secure
                                            if true, uses SSL/TLS to communicate with the domain
                                           application server.
                                            file containing the domain application server password.
                 --passwordfile
                                           indicates that any output data must be very concise,
                 --terse
                                            typically avoiding human-friendly sentences and favoring
                                            well-formatted data for consumption by a script.
                                           setting to true will echo the command line statement on
                 --echo
                                            the standard output.
                 --interactive
                                            prompts you for the required options that are not already
                                            specified.
                                           name of the JDBC connection pool. If two or more JDBC
                 --connectionpoolid
                                            resource elements point to the same connection pool
                                            element, they will use the same pool connections at
                                            runtime.
                 --enabled
                                            determines whether the resource is enabled at runtime.
                 --description
                                            text description of the JDBC resource.
  OPERANDS
                              JNDI name for resource to be created.
                 jndi_name
   EXAMPLES
                 EXAMPLE 1 Using the create-jdbc-resource command
                 asadmin> create-jdbc-resource --user admin1
                 --password adminadmin1 --host pigeon --port 5001
                 --connectionpoolid XA connection pool --description
                 "creating a sample jdbc resource" sample_jdbc_resource
                 Command create-jdbc-resource executed successfully
```

create-jdbc-resource – registers the JDBC resource

NAME

create-jdbc-resource(1)

EXAMPLE 1 Using the create-jdbc-resource command (Continued)

Where: sample_jdbc_resource is the resource that is created.

EXIT STATUS

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

SEE ALSO

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

SYNOPSIS	create-jmsdestuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive]desttype -T topic queue [property (name=value)[:name=value]*] dest_name	
DESCRIPTION	creates the JMS physical destination. Along with the physical destination, you create a JMS destination resource with a Name property that specifies the physical destination. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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 re specified.		prompts you for the required options that are not already specified.
	desttype	type of JMS destination. Valid values are topic, and queue.
	property	name/value pairs used for specifying MQ specific attributes to further customize the destination to be created.
OPERANDS	dest_name name of the JMS destination. Valid value is any name that can be a Java identifier.	
EXAMPLES	EXAMPLE 1 Using create-jmsdest	
	asadmin> create-jmsdestuser adminpasswordfile passwords.txthost localhostport 4848desttype queueproperty User=public:Password=public PhysicalQueue Command create-jmsdest executed successfully	
EXIT STATUS	0 command executed successfully	

 $\pmb{NAME} \hspace{0.1cm} | \hspace{0.1cm} create\text{-}jmsdest - adds \hspace{0.1cm} the \hspace{0.1cm} physical \hspace{0.1cm} destination$

create-jmsdest(1)

error in executing the command

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

create-jmsobj(1)

NAME | create-jmsobj – adds the named object **SYNOPSIS** create-jmsobj --instance instance_name --jndilookupname | -1 jndi_lookup_name --objtype | -o object_ty **DESCRIPTION** Adds the named destination. **OPTIONS** --instance the name of the instance. --jndilookupname --objtype --jmsobjattr Lists the optional attribute=value pairs for configuring the resource. **EXAMPLES** asadmin% create-jmsobj --instance server1 **INTERFACE** JMS Destination pages **EQUIVALENT SEE ALSO** delete-jmsobj(1) list-jmsobj(1)

create-jms-resource(1)

)(-	-/		
NAME	create-jms-resource – registers the JMS resource		
SYNOPSIS	<pre>create-jms-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]restype resource_type [enabled=true] [description text] [property (name=value) [:name=value] *] jndi_name</pre>		
DESCRIPTION	Registers the JMS resource. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
	restype	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.	
	description	text description of the JMS resource.	
	property	optional attribute name/value pairs for configuring the JMS resource.	
OPERANDS	jndi_name JNDI na	ame of the JMS resource to be created.	
EXAMPLES	EXAMPLE 1 Creating a JMS connection factory resource for durable subscriptions asadmin> create-jms-resourceuser admin1password adminadmin1host pigeonport 5001restype javax.jms.TopicConnectionFactorydescription "example of creating a JMS connection factory"		

EXAMPLE 1 Creating a JMS connection factory resource for durable subscriptions (Continued)

--property ClientId=MyID jms/DurableTopicConnectionFactory

Command create-jms-resource executed successfully

Where: jms/DurableTopicConnectionFactory is the JNDI name of the resource, and the ClientId property sets a client ID on the connection factory so that it can be used for durable subsciptions. The JNDI name for a JMS resource customarily includes the jms/ naming subcontext.

EXAMPLE 2 Creating a JMS destination resource

```
asadmin> create-jms-resource --user admin1
--password adminadmin1 --host pigeon --port 5001
--restype javax.jms.Queue
--property Name=PhysicalQueue jms/MyQueue
Command create-jms-resource executed successfully
```

Where: jms/Queue is the JNDI name of the resource, and the Name property specifies the physical destination that the resource refers to.

EXIT STATUS

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

SEE ALSO

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

create-jndi-resource(1)

NAME	create-jndi-resource – registers the JNDI resource		
SYNOPSIS	<pre>create-jndi-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] jndilookupname lookup_namerestype typefactoryclass class_name [enabled= [description text] [property (name=value) [:name=value]*] jndi_name</pre>		
DESCRIPTION	Registers the JNDI resource. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
	jndilookupname	lookup name used by external container.	
	restype	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.	
	description	text description of the JDBC connection pool.	
	property	optional attribute name/value pairs for configuring the JNDI resource.	
OPERANDS	jndi_name name of th	e JNDI resource to be created.	

EXAMPLES | **EXAMPLE 1** Using the create-jndi-resource command

asadmin> create-jndi-resource --user admin1

- --password adminadmin1 --host pigeon --port 5001
- --jndilookupname sample_jndi --restype queue --factoryclass sampleClass
- --description "this is a sample jndi resource" sample_jndi_resource

Command create-jndi-resource executed successfully

Where: sample_jndi_resource is the JNDI resource created.

EXIT STATUS

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

SEE ALSO

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

create-jvm-options(1)

Create-jvm-options - creates the JVM options from the Java configuration or profiler elements Create-jvm-optionsuser admin_user	eate-jvm-options(1)		
C-password admin_password] [host localhost] [port 4848] [secure] -s] [passwordfile filename] [terse=false] [echo-false] [interactive=true] [profiler=false] [-ymoption_name=jvm_option_value] [:jvm_option_name=jvm_option_value] [:jvm_option_name=jvm_option_value]	NAME			
domain.xml file. You can enter more than one JVM option separated by a colon (;). If the JVM option starts with a dash (-) then use two dashes (—) before the operand to distinguish that JVM option is an operand and not an option. JVM options are used to record the settings needed to get a particular profiler going. You must restart the server for the newly created JVM options to take affect. Use the start-domain command to restart the server domain. OPTIONS user authorized domain application server administrative username. password password to administer the domain application server. host machine name where the domain application server is running. port port number of the domain application server listening for administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS OPERANDS OPERANDS Jum_option_name=jum_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jum_option_value. Additionally, you can use ":" as a delimiter for more than one	SYNOPSIS	[password & [secure -s [echo=fals] (jvm_option	admin_password] [password e] [inter _name=jvm_o	[][host localhost] [port 4848] rdfile filename] [terse=false] active=true] [profiler=false ption_value)
OPTIONS user authorized domain application server administrative username. password password to administer the domain application server. host machine name where the domain application server is running. port port number of the domain application server listening for administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one	DESCRIPTION	domain.xml file. You can enter more than one JVM option separated by a colon (:) . If the JVM option starts with a dash (-) then use two dashes (—) before the operand to distinguish that JVM option is an operand and not an option. JVM options are used to		
username. password password to administer the domain application server. host machine name where the domain application server is running. port port number of the domain application server listening for administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS Jom_option_name=jom_option_value the left side of the equal sign (=) is the jVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one				
host machine name where the domain application server is running. port port number of the domain application server listening for administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one	OPTIONS	user		omain application server administrative
port port number of the domain application server listening for administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		password	password to a	administer the domain application server.
administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		host	machine nam	e where the domain application server is running.
application server. passwordfile file containing the domain application server password. 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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS OPERANDS image: promption_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		port	-	
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. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		secure		
avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. echo setting to true will echo the command line statement on the standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		passwordfile	file containing	g the domain application server password.
standard output. Default is false. interactive if set to true (default), only the required password options are prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		terse	avoiding hum well-formatte	nan-friendly sentences and favoring
prompted. profiler indicates if the JVM options is for the profiler. Profiler must exist for this option to be true. OPERANDS jvm_option_name=jvm_option_value the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		echo		
operands operan		interactive		default), only the required password options are
option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one		profiler		
	OPERANDS	jvm_option_name=jvm_	option_value	option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one

the escape character \ to offset the ":" delimiter.

EXAMPLES

EXAMPLE 1 Using create-jvm-options

```
asadmin> create-jvm-options --user admin --password adminadmin
--host localhost --port 4848 --profiler=false --DDebug=true:"-Xmx256m":"
-Dcom.sun.aas.imqBin"="\/export\/as7se\/imq\/bin"
Command create-jvm-options executed successfully
```

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

EXIT STATUS

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

SEE ALSO

delete-jvm-options(1)

create-lifecycle-module(1)

NAME	create-lifecycle-module – adds a lifecycle module		
SYNOPSIS	create-lifecycle-moduleuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]classname class_name [classpath classpath] [loadorder load_order] [failurefatal=false] [enabled=true] [description description] [property (name=value) [:name=value] *] module_name		
DESCRIPTION	Creates the lifecycle. T	his command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	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.	
	enabled	determines whether the resource is enabled at runtime.	
	description	text description of the resource.	

create-lifecycle-module(1)

optional attribute name/value pairs for configuring the --property

resource.

OPERANDS

unique identifier for the deployed server lifecycle event listener module_name

module.

EXAMPLES

EXAMPLE 1 using create-lifecycle-module

asadmin> create-lifecycle-module --user admin --password adminadmin

- --host fuyako --port 7070 --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

Command create-lifecycle-module executed successfully

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)

create-mime(1)

NAME

create-mime – adds the MIME type

SYNOPSIS

create-mime

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

DESCRIPTION

Adds the MIME type associated with the named server instance. The server determines the MIME type of a requested resource by invoking the type-by-extension directive in the <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.

--mimefile name of a MIME types file.

OPERANDS

mime_id internal name for the MIME types listing. It is used in a virtual-server

element to define the MIME types used by the virtual server.

EXAMPLES

EXAMPLE 1 Using create-mime

asadmin> create-mime --user admin --password adminadmin --host fuyako --port 7070 --instance server Created Mime with id = sampleMIME

Where: sampleMIME is the name of the MIME created.

EXIT STATUS

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

INTERFACE EQUIVALENT SEE ALSO

HTTP Server node, MIME Type Files page

delete-mime(1), list-mimes(1)

SYNOPSIS	create-persistence-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] [jdbcjndiname jdbc_jndiname connectionpoolid id] [factoryclass classname] [enabled=true] [description text] [property (name=value) [:name=value] *] jndi_name	
DESCRIPTION		source. This command is supported in remote mode only.
	Thejdbcjndiname opt exclusive; only one should	tion andconnectionpoolid option are mutually be used.
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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. Default is false.
	echo	setting to true will echo the command line statement on the standard output. Default is false.
	interactive	if set to true (default), only the required password options are prompted.
	jdbcjndiname	JDBC resource with which database connections are obtained. Must be the name of one of the pre-created JDBC resources.
	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.
	factoryclass	class that creates persistence manager instance.

 $\pmb{NAME} \mid create\text{-persistence-resource} - registers the persistence resource$

create-persistence-resource(1)

--enabled determines whether the resource is enabled at runtime or

not.

--description text description of the persistence resource.

--property optional attribute name/value pairs for configuring the

persistence resource.

OPERANDS | jndi_nan

jndi_name JNDI name of the persistence resource.

EXAMPLES

EXAMPLE 1 Using create-persistence-resource

asadmin> create-persistence-resource --user admin1 --password adminadmin1
--jdbcjndiname sample_jndi_resource --factoryclass "com.pmf.class"
sample_persistence_resource

Command create-persistence-resource executed successfully

Where: sample_persistence_resource is the persistence manager factory resource created.

EXIT STATUS

- command executed successfully
- 1 error in executing the command

SEE ALSO

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

NAME | create-profiler – creates the profiler element

SYNOPSIS

create-profiler --user admin_user [--password admin_password]

[--host localhost] [--port 4848] [--secure|-s]

[--passwordfile filename] [--terse=false] [--echo=false]

[--interactive=true] [--classpath classpath]

[--nativelibpath native_library_path] [--enabled=true]

[--property (name=value) [:name=value] *] profiler_name

DESCRIPTION

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

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
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.

create-profiler(1)

OPERANDS

| *profiler_name* name of the profiler.

EXAMPLES

EXAMPLE 1 Using create-profiler

asadmin> create-profiler --user admin --passwordfile passwords.txt
--host localhost --port 4848 --classpath com.iplanet.ias.profile.Profiler
--nativelibpath /u/home/lib --enabled=false
--property defaultuser=admin:password=adminadmin sample_profiler
Created Profiler with id = sample profiler

Where: sample profiler is the profiler created.

EXIT STATUS

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

SEE ALSO

delete-profiler(1)

NAME

create-resource-adapter-config - creates the configuration information in domain.xml for the connector module

SYNOPSIS

create-resource-adapter-config --user admin_user

[--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--threadpools threadpoolids] [--property name=value[:name=value]*] raname

DESCRIPTION

Creates configuration information for the connector module. This command can be executed prior to deploying a resource adapter, so that the configuration information is available at the time of deployment, or after deployment. If the resource adapter is created after deployment, the resource adapter is started. You must first create a threadpool, using the create-threadpool command, and then identify that threadpool value as the ID in the--threadpools option.

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
threadpool	the threadpool ID from which the work manager gets the

OPERANDS

--property

raname the value kept in the resource-adapter-name in the domain.xml file.

configuration properties of the resource adapter java bean.

EXAMPLES

EXAMPLE 1 Using create-resource-adapter-config

thread.

asadmin> create-resource-adapter-config --username admin1 --password adminadmin myresourceadapter

create-resource-adapter-config(1)

EXAMPLE 1 Using create-resource-adapter-config (Continued)

Command create-resource-adapter-config executed successfully

EXIT STATUS

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

SEE ALSO

create-threadpool(1), delete-resource-adapter-config(1),
list-resource-adapter-configs(1)

NAME | create-ssl – Creates the SSL element in the HTTP listener, IIOP listener, or IIOP Service **SYNOPSIS** create-ssl --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --type [http-listener|iiop-listener|iiop-service] --certname cert_name [--ssl2enabled=false] [--ssl2ciphers ssl_2_ciphers] [--ssl3enabled=true] [--ssl3tlsciphers ssl3_tls_ciphers] [--tlsenabled=true]

DESCRIPTION

Creates the ssl element from the HTTP listener, IIOP listener, or IIOP service. The *listener_id* is not required if the --type option is iiop-service.

[--tlsrollbackenabled=true] [--clientauthenabled=false]

This command is supported in remote mode only.

[listener_id]

OPTIONS

5	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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. Default is false.
	echo	setting to true will echo the command line statement on the standard output. Default is false.
	interactive	if set to true (default), only the required password options are prompted.
	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

create-ssl(1)

		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 are: rsa_rc4_128_md5, rsa3des_sha, rsa_des_sha, rsa_rc4_40_md5, rsa_rc2_40_md5, rsa_nul1_md5. Allowed TSL values are: rsa_des_56_sha, rsa_rc4_56_sha. If no value is specified, all supported ciphers are assumed to be enabled.	
	tlsenabled	determines whether TLS is enabled.	
	tlsrollbackenabled	determines whether TLS rollback is enabled. TLS rollback should be enabled for Microsoft Internet Explorer 5.0 and 5.5.	
	clientauthenabled	determines whether SSL3 client authentication is performed on every request independent of ACL-based access control.	
,	listener_ID the ID of the listener or service that the SSL is created for. This operand is not required if thetype option is iiop-service.		
ì	EXAMPLE 1 Using create-ssl		
	asadmin> create-ssluser adminpassword adminadminhost fuyakoport 7070type http-listenercertname sampleCertssl2enabled=truessl2ciphers rc4,rc2,desssl3enabled=falsessl3tlscipers rsa_rc4_128_md5,rsa3des_sha,rsa_des_sha, rsa_rc4_40_md5tlsenabled=falsetlsrollbackenabled=falseclientauthenabled=false http-listener-1 Created SSL in HTTP Listener		
	Where: SSL is created for http-listener-1.		
,	0 command executed successfully		
	1 error in executing the cor	mmand	
)	delete-ssl(1)		

EXIT STATUS

OPERANDS

EXAMPLES

NAME | create-threadpool – adds a threadpool

SYNOPSIS

create-threadpool --user admin_user

[--password admin_password] [--host localhost]

[--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true]

[--maxthreadpoolsize max_thread_pool_size] [--minthreadpoolsize min_thread_pool_size] [--idletimeout idle_thread_timeout_in_seconds]

[--workqueues number_work_queues] threadpool_id

DESCRIPTION

Creates a thread-pool with the specified name. You can specify maximum and minimum number of threads in the pool, the number of work queues, and the idle timeout of a thread. The created thread pool can be used for servicing IIOP requests and for resource adapters to service work management requests. Please note that a created thread pool can be used in multiple resource adapters. This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
maxthreadpoolsize	maximum number of threads in the threadpool servicing requests in this queue. This is the upper bound on the number of threads that exist in the threadpool.
minthreadpoolsize	minimum number of threads in the threadpool servicing requests in this queue. These are created up front when the threadpool is instantiated.
idletimeout	idle threads are removed from the pool after this time.

create-threadpool(1)

--workqueues identifies the total number of work queues serviced by

this threadpool.

OPERANDS | threadpool_id an ID for the work queue; for example, thread-pool-1, thread-pool-2,

etc.

EXAMPLES | **EXAMPLE 1** Using create-threadpool

asadmin> create-threadpool --user admin1 --password adminadmin1

--maxthreadpoolsize 100 --minthreadpoolsize 20 --idletimeout 2 --workqueues 100 threadpool-1 $\,$

Command create-threadpool executed successfully

EXIT STATUS 0 command executed successfully

1 error in executing the command

SEE ALSO | delete-threadpool(1), list-threadpools(1)

NAME | create-virtual-server – adds the named virtual server

SYNOPSIS

create-virtual-server --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --hosts hosts [--httplisteners http_listeners] [--defaultwebmodule default_web_module] [--state on] [--logfile log_file] [--property (name=value) [:name=value] *] virtual_server_id

DESCRIPTION

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

This command is supported in remote mode only.

OPTIONS --user

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
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.
httplisteners	optional; a comma separated (,) list of HTTP listener IDs. Required only for a virtual server that is not the default virtual server.

create-virtual-server(1)

--defaultwebmodule standalone web module associated with this virtual

server by default.

--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.

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

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

connection pool.

OPERANDS *virtual_server_id*

identifies the unique ID for the virtual server to be created. This

virtual server ID cannot begin with a number.

EXAMPLES | **EXAMPLE 1** Using create-virtual-server

asadmin> create-virtual-server --user admin1 --password adminadmin1

--host pigeon --port 5001 --httplisteners http-listener-1 --defaultwebmodule simple

--state on --logfile server.log --property ${\tt User=admin:Password=admin}$

--hosts sample1,sample2 sample_vs1

Command create-virtual-server executed successfully

Where sample vs1 is the virtual server created.

EXIT STATUS 0 command executed successfully

1 error in executing the command

SEE ALSO | delete-virtual-server(1), list-virtual-server(1)

NAME | delete-acl – removes the access control list file

SYNOPSIS

delete-acl

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

DESCRIPTION

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

OPTIONS

--user administrative user associated for the instance.

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

--secure indicates communication with the administrative instance in

secured mode.

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

administrative instance).

--instance name of the instance.

OPERANDS

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

EXAMPLES

EXAMPLE 1 Using delete-acl

asadmin> delete-acl --user admin --password adminadmin --host fuyako --port 7070 --instance server Deleted ACL with id = sampleACL

Where: sampleACL is the ACL that is deleted.

EXIT STATUS

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

INTERFACE EQUIVALENT SEE ALSO Access Control List page

create-acl(1), list-acl(1)

delete-admin-object(1)

NAME	delete-admin-object – removes the administered object with the specified JNDI name		
SYNOPSIS	delete-admin-objectuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] jndi_name		
DESCRIPTION	Removes the administered object with the specified JNDI name. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	jndi_name JNDI name of the administered object to be deleted.		
EXAMPLES	EXAMPLE 1 Using delete-admin-object		
	The example listed in the add-admin-object command should be executed before attempting to execute this example:		
	asadmin> delete-admin-objectsuser adminpassword admin123 jms/samplequeue		
	Deletes the previously created administration object with the JNDI name jms/sample.		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-admin-object(1), list-admin-objects(1)		

SYNOPSIS delete-audit-module --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] audit_module_name **DESCRIPTION** Removes the named audit module. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. machine name where the domain application server is running. --host port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. name of the audit module to be deleted. **OPERANDS** audit_module_name **EXAMPLES EXAMPLE 1** Using delete-audit-module asadmin> delete-audit-module --user admin1 --password adminadmin1 --host pigeon --port 5001 sampleAuditModule Command delete-audit-module executed successfully **EXIT STATUS** 0 command executed successfully 1 error in executing the command **SEE ALSO** create-audit-module(1), list-audit-modules(1)

create-audit-module - removes the named audit-module

NAME

delete-authdb(1)

NAME | delete-authdb - removes the authorized database **SYNOPSIS** delete-authdb --user admin_user[--password admin_password][--host localhost] [--port 4848] [--passwordfile filename] [--secure | -s] [--instance instance_name] --virtualserver virtualserver_ID authdb_ID Removes the authorized database associated with the named server instance. **DESCRIPTION OPTIONS** --user administrative user associated for the instance. administrative password corresponding to the administrative --password --host host name of the machine hosting the administrative instance.

> administrative port number associated with the --port

administrative host.

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

administrative instance).

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

instance.

name of the instance. --instance

--virtualserver virtual server ID. It can also be referred to as the variable \$id

in an obj.conf file. A virtual server ID cannot begin with a

number.

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

EXAMPLES EXAMPLE 1 Using delete-authdb

> asadmin> delete-authdb --user admin --password adminadmin --host fuyako --port 7070 --instance serv Deleted AuthDB with id = sampleAuth

Where: sampleAuth is the authob deleted.

EXIT STATUS command executed successfully

> 1 error in executing the command

SEE ALSO create-authdb(1), list-authdbs(1)

1471111	defete duti fedini Tentoves the fidined dutionized fedini		
SYNOPSIS	<pre>delete-auth-realmuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] auth_realm_name</pre>		
DESCRIPTION	Removes the named a only.	authorized realm. This command is supported in remote mode	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	auth_realm_name r	name of this realm.	
EXAMPLES	EXAMPLE 1 Using delete	e-auth-realm	
	asadmin> delete-auth-realmuser admin1passwordfile password		
	host pigeonport 5001 db		
	Command delete-auth-realm executed successfully		
	Where db is the auth realm deleted.		
EXIT STATUS	0 command executed successfully		
	1 error in executing the command		
SEE ALSO	create-auth-real	m(1), list-auth-realms(1)	

 $\textbf{NAME} \hspace{0.1cm}|\hspace{0.1cm} \textbf{delete-auth-realm-removes the named authorized realm}$

delete-connector-connection-pool(1)

NAME	delete-connectior-connection-pool – removes the specified connection pool		
SYNOPSIS	delete-connector-connection-pooluser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] [cascade=false] pool_name		
DESCRIPTION	removes the specified only.	connection pool. This command is supported in remote mode	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	cascade	if set to to true, the pool and all the connector resources associated with the pool are deleted. If set to false (default), the deletion of the pool fails if there are any resources associated with the pool. Either the resource has to be deleted explicitly or the option must be set to true.	
OPERANDS	pool_name name of	the connection pool to be deleted.	
EXAMPLES	EXAMPLE 1 Using delete-connector-connection-pool		
	Execute the example described for create-connector-connection-pool command before trying out this example:		
	asadmin> delete-connector-connection-pooluser adminpassword adminadmin jms/qConnPool Command delete-connector-connection-pool executed successfully		

delete-connector-connection-pool(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command

SEE ALSO

create-connector-connection-pool(1), list-connector-connection-pools(1)

delete-connector-resource(1)

NAME	delete-connector-reso	urce – removes the named connector resource	
SYNOPSIS	delete-connector-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] connector_resource_name		
DESCRIPTION	Removes the named resource. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	connector_resource_nar	ne name of the connector resource to be deleted.	
EXAMPLES	EXAMPLE 1 Using the de	elete-connector-resource command	
	Execute the example described for create-connector-resource command before trying out this example: asadmin> delete-connector-resourceuser admin1password adminadmin1host pigeonport 5001 jms/qConnFactory Command delete-connector-resource executed successfully		
	This will delete the p	reviously created connector resource jms/qConnPool.	
EXIT STATUS	0 command execu	ated successfully	
	1 error in executin	ng the command	
SEE ALSO	create-connector	r-resource(1), list-connector-resources(1)	

NAME

delete-connector-security-map – deletes the named security map for the given connector connection pool

SYNOPSIS

delete-connector-security-map --user admin_user

[--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] --poolname

connector_connection_pool_name mapname

DESCRIPTION

Deletes the named security map for the given connector connection pool. This command is supported in remote mode only.

OPTIONS

user	authorized	domain	application	server administrative

username.

--password password to administer the domain application server.

--host machine name where the domain application server is running.

--port port number of the domain application server listening for

administration requests.

--secure if true, uses SSL/TLS to communicate with the domain

application server.

--passwordfile file containing the domain application server password.

--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. Default is

false.

--echo setting to true will echo the command line statement on the

standard output. Default is false.

--interactive if set to true (default), only the required password options are

prompted.

--poolname connector connection pool name for which the security map

that is to be deleted belongs to.

OPERANDS

mapname name of the security map to be deleted.

EXAMPLES

EXAMPLE 1 Using delete-connector-security-map

asadmin> delete-connector-security-map --user admin --password adminadmin
--poolname connector-pool1 securityMap1

Command delete-connector-security-map executed successfully

EXIT STATUS

- command executed successfully
- 1 error in executing the command

delete-connector-security-map(1)



SYNOPSIS delete-custom-resource --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] jndi_name DESCRIPTION Removes the custom resource. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **OPERANDS** jndi_name JNDI name of the custom resource to be deleted. **EXAMPLES EXAMPLE 1** Using delete-custom-resource asadmin> delete-custom-resource --user admin1 --password adminadmin1 --host pigeon --port 5001 sample custom resource Command delete-custom-resource executed successfully Where sample custom resource is the custom resource deleted. **EXIT STATUS** 0 command executed successfully error in executing the command **SEE ALSO** create-custom-resource(1), list-custom-resources(1)

delete-custom-resource – removes the custom resource

NAME

delete-domain(1)

erete domain(1)			
NAME	delete-domain – deletes the given domain		
SYNOPSIS	delete-domain [terse=false] [echo=false] [domaindir install_dir/domains] domain_name		
DESCRIPTION	delete-domain deletes the specified domain. The domain must already exist and must be stopped. This command is supported in local mode only.		
OPTIONS	domaindir	directory where the domain is to be deleted. If specified, path must be accessible in the filesystem. If not specified, the domain in the default <code>install_dir/domains</code> directory is deleted.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
OPERANDS	domain_name r	name of the domain. Must be a unique name.	
EXAMPLES	EXAMPLE 1 Using delete-domain		
	asadmin> delete-	domain sampleDomain	
	deleted domain sa	ampleDomain successfully	
	Where: the sampleDomain domain is deleted.		
EXIT STATUS	0 command ex	xecuted successfully	
	1 error in exec	cuting the command	
SEE ALSO	create-domain	(1), start-domain(1), stop-domain(1), list-domains(1)	

NAME

delete-file-user – removes the named file user

SYNOPSIS

delete-file-user --user admin_user [--password admin_password]
 [--host localhost] [--port 4848] [--secure|-s]

[--passwordfile filename] [--terse=false] [--echo=false]

[--interactive=true] [--authrealmname auth_realm_name] user_name

DESCRIPTION

Deletes the entry in the keyfile with the specified username. If the <code>auth_realm_name</code> is not specified, an entry is created in the default keyfile. If <code>auth_realm_name</code> is specified, an entry is created in the keyfile where the <code>auth-realm</code> name in the <code>domain.xml</code> file points to.

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative
	username.

--password password to administer the domain application server.

--host machine name where the domain application server is

running.

--port port number of the domain application server listening for

administration requests.

--secure if true, uses SSL/TLS to communicate with the domain

application server.

--passwordfile file containing the domain application server password.

--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. Default is

false

--echo setting to true will echo the command line statement on the

standard output. Default is false.

--interactive if set to true (default), only the required password options are

prompted.

--authrealmname name, in the domain.xml file, where you have different

stores for file auth realm.

OPERANDS

user_name name of file user to be deleted.

EXAMPLES

EXAMPLE 1 Using the delete-file-user command

It is assumed that an authority realm has already been created using the create-auth-realm command.

```
asadmin> delete-file-user --user admin1 --password adminadmin1 --host pigeon --port 5001 --authrealmname auth-realm1 sample_user Command delete-file-user executed successfully
```

delete-file-user(1)

EXAMPLE 1 Using the delete-file-user command (*Continued*)

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)

		defete http hotelier(1)	
NAME	delete-http-listener – removes the HTTP listener		
SYNOPSIS	<pre>delete-http-listeneruser admin_user[password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] httplistener_ID</pre>		
DESCRIPTION	Removes the HTTP li	steners. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	listener_id listener	ID of the HTTP listener.	
EXAMPLES	EXAMPLE 1 Using delete	e-http-listener	
	asadmin> delete-http-listeneruser admin1password adminadmin1host pigeonport 5001 sampleListener Deleted http listener with id = sampleListener Where: sampleListener is the HTTP listener deleted.		
EXIT STATUS	0 command execu	ited successfully	
	1 error in executir	ng the command	
SEE ALSO	create-http-list	ener(1), list-http-listeners(1)	

delete-http-qos(1)

NAME delete-http-qos - removes the quality of service parameter **SYNOPSIS** delete-http-qos --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--passwordfile filename] [--secure | -s] [--virtualserver virtual_server_ID] instance_name DESCRIPTION Removes the quality of service pamameter associated with the named server instance. **OPTIONS** --user administrative user associated for the instance. administrative password corresponding to the administrative --password user. host name of the machine hosting the administrative instance. --host administrative port number associated with the --port administrative host. --passwordfile file containing passwords appropriate for the command (e.g., administrative instance). if true, uses SSL/TLS to communicate with the administrative --secure instance. virtual server ID. It can also be referred to as the variable \$id --virtualserver in an obj.conf file. A virtual server ID cannot begin with a number. **OPERANDS** instance_name name of the instance. **EXAMPLES EXAMPLE 1** Using delete-http-qos asadmin> delete-http-qos --user admin --password adminadmin --host fuyako --port 7070 --virtualserv 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 error in executing the command 1 **INTERFACE** Server instance, HTTP Server Virtual Servers, Instance QOS page for the server **EQUIVALENT** instance **SEE ALSO** create-http-qos(1)

```
SYNOPSIS
                 delete-iiop-listener --user admin_user [--password admin_password]
                       [--host localhost] [--port 4848] [--secure | -s]
                       [--passwordfile filename] [--terse=false ] [--echo=false ]
                       [--interactive=true ] listener_ID
DESCRIPTION
                 Removes the IIOP listener. This command is supported in remote mode only.
    OPTIONS
                 --user
                                       authorized domain application server administrative
                                       username.
                  --password
                                       password to administer the domain application server.
                 --host
                                       machine name where the domain application server is running.
                                       port number of the domain application server listening for
                 --port
                                       administration requests.
                 --secure
                                       if true, uses SSL/TLS to communicate with the domain
                                       application server.
                 --passwordfile
                                       file containing the domain application server password.
                                       indicates that any output data must be very concise, typically
                 --terse
                                       avoiding human-friendly sentences and favoring
                                       well-formatted data for consumption by a script. Default is
                                       false.
                 --echo
                                       setting to true will echo the command line statement on the
                                       standard output. Default is false.
                 --interactive
                                       if set to true (default), only the required password options are
                                       prompted.
  OPERANDS
                 listener_id
                              unique identifier for the IIOP listener to be deleted.
   EXAMPLES
                 EXAMPLE 1 Using delete-iiop-listener
                 asadmin> delete-iiop-listener --user admin --password adminadmin
                 --host fuyako --port 7070 sample iiop listener
                 Command delete-iiop-listener executed successfully
                 Where: sample iiop listener is the IIOP listener deleted.
 EXIT STATUS
                      command executed successfully
                 1
                      error in executing the command
    SEE ALSO
                 create-iiop-listener(1), list-iiop-listeners(1)
```

delete-iiop-listener – removes the IIOP listener

NAME

delete-instance(1)

NAME | delete-

delete-instance – deletes the instance that is not running.

SYNOPSIS

delete-instance [--user admin_user] [--password admin_password]

[--host localhost] [--port 4848] [--local=false] [--domain domain_name] [--passwordfile filename]

[--secure|-s] instance_name

DESCRIPTION

Use the delete-instance command to delete the instance that you specify. The delete-instance command can be run both locally and remotely. To delete the instance locally, not requiring the administration server to be up and running, specify the --local option. To delete the instance remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server. Use this command with discretion since it is destructive and there is no undo.

OPTIONS

	1	
user	administrative user associated for the instance.	

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

--domain name of the domain.

--local determines if the command should delegate the request to

administrative instance or run locally.

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

administrative instance).

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

instance.

OPERANDS

instance_name name of the instance to be deleted.

EXAMPLES

EXAMPLE 1 Using delete-instance in local mode

asadmin> delete-instance --domain domain1 server1

Deleted Instance server1 successfully

Where: the server1 instance for the domain1 domain is deleted on the local machine.

EXAMPLE 2 Using delete-instance in remote mode

asadmin> delete-instance --user admin --passwordfile passwords.txt --host localhost --port 4848 ser Deleted Instance server1 successfully **EXAMPLE 2** Using delete-instance in remote mode (Continued)

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

- command executed successfully
- 1 error in executing the command

INTERFACE EQUIVALENT SEE ALSO

Server Instance page

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

delete-javamail-resource(1)

NAME	delete-javamail-resource – removes the JavaMail resource		
SYNOPSIS	<pre>delete-javamail-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] jndi_name</pre>		
DESCRIPTION	Removes the JavaMai	l resource. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
OPERANDS	jndi_name JNDI na	nme of the JavaMail resource to be deleted.	
EXAMPLES	EXAMPLE 1 Using delete-javamail-resource		
	asadmin> delete-javamail-resourceuser adminpassword adminadminhost fuyakoport 7070 mail/MyMailSession Command delete-javamail-resource executed successfully Where: mail/MyMailSession is the JavaMail resource deleted.		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	•	
SEE ALSO		resource(1), list-javamail-resources(1)	

SYNOPSIS delete-jdbc-connection-pool --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive] [--cascade=false] connection_pool_ID **DESCRIPTION** Removes the JDBC connection pool. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. file containing the domain application server password. --passwordfile indicates that any output data must be very concise, typically --terse 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. --cascade If set to true, deletes all the connector resources associated with the pool, apart from the pool itself. If set to false (default), the deletion of the pool fails if there are any resources associated with the pool. Either the resource has to be deleted explicitly, or set the option to true. **OPERANDS** connection_pool_id name of the JDBC connection pool to be deleted. **EXAMPLES EXAMPLE 1** Using the delete-jdbc-connection-pool command asadmin> delete-jdbc-connection-pool --user admin --password adminadmin --host fuyako port 7070 XA_connection_pool Deleted the JDBC connection pool resource with id = XA connection pool Where: the XA connection pool resource is deleted. **EXIT STATUS** 0 command executed successfully 1 error in executing the command

delete-jdbc-connection-pool – removes the JDBC connection pool

NAME

delete-jdbc-connection	on-pool(1)
SEE ALSO	<pre>create-jdbc-connection-pool(1), list-jdbc-connection-pools(1)</pre>

```
SYNOPSIS
                 delete-jdbc-resource --user admin_user [--password admin_password]
                       [--host localhost] [--port 4848] [--secure | -s]
                       [--passwordfile filename] [--terse=false] [--echo=false]
                       [--interactive=true] jndi_name
DESCRIPTION
                 Removes the JDBC resource. This command is supported in remote mode only.
    OPTIONS
                 --user
                                      authorized domain application server administrative
                                      username.
                 --password
                                      password to administer the domain application server.
                 --host
                                      machine name where the domain application server is running.
                                      port number of the domain application server listening for
                 --port
                                      administration requests.
                 --secure
                                      if true, uses SSL/TLS to communicate with the domain
                                      application server.
                 --passwordfile
                                      file containing the domain application server password.
                                      indicates that any output data must be very concise, typically
                 --terse
                                      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.
  OPERANDS
                              name of the JDBC resource to be deleted.
                 jndi_name
   EXAMPLES
                 EXAMPLE 1 Using the delete-jdbc-resource command
                 asadmin> delete-jdbc-resource --user admin1
                 --password adminadmin1 --host pigeon --port 5001
                 sample_jdbc_resource
                 Command delete-jdbc-resource executed successfully
                 Where: sample jdbc resource is the resource that is deleted.
 EXIT STATUS
                 0
                      command executed successfully
                      error in executing the command
    SEE ALSO
                 create-jdbc-resource(1), list-jdbc-resources(1)
```

delete-jdbc-resource – removes the JDBC resource

NAME

delete-jmsdest(1)

NAME	delete-jmsdest – destr	roys the physical destination	
SYNOPSIS	<pre>delete-jmsdestuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive]desttype -T topic queue dest_name</pre>		
DESCRIPTION	Destroys the physical destinations. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
	desttype	type of JMS destination. Valid values are topic, and queue.	
OPERANDS	dest_name name of the JMS destination. Valid value is any name that can be a Java identifier.		
EXAMPLES	EXAMPLE 1 Using dele	te-imsdest	
	asadmin> delete-jmsdestuser adminpassword adminadminhost localhostport 4848desttype queue PhysicalQueue Command delete-jmsdest executed successfully		
EXIT STATUS	0 command execu	ated successfully	
	1 error in executir	•	
SEE ALSO	create-jmsdest(1)	,list-jmsdest(1)	

delete-jmsobj – destroys the named object NAME |

SYNOPSIS delete-jmsobj

--instance instance_name --jndilookupname | -1 jndi_lookup_name

DESCRIPTION Destroys the named destinations.

> **OPTIONS** --instance the name of the instance.

> > --jndilookupname

EXAMPLES asadmin% delete-jmsobj --instance server1 --jndilookup

INTERFACE EQUIVALENT

unknown

SEE ALSO create-jmsobj(1) list-jmsobj(1)

delete-jms-resource(1)

NAME	delete-jms-resource –	removes the JMS resource	
SYNOPSIS	delete-jms-resourceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] jndi_name		
DESCRIPTION	Removes the JMS resource. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
OPERANDS	jndi_name JNDI na	ame of the JMS resource to be deleted.	
EXAMPLES	EXAMPLE 1 Using the delete-jms-resource command		
	asadmin> delete-jms-resourceuser admin1password adminadmin1		
	host pigeonport 5001 sample_jms_resource Command delete-jms-resource executed successfully		
	Where: sample_jms	_resource is the resource that is deleted.	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	, and the second	
SEE ALSO		rce(1), list-jms-resources(1)	
	,		

SYNOPSIS	delete-jndi-resourceuser admin_user[password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] jndi_name		
DESCRIPTION	Removes the JNDI res	source. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
OPERANDS	<i>jndi_name</i> name of	f the JNDI resource to be deleted.	
EXAMPLES	EXAMPLE 1 Using the delete-indi-resource command		
		-resourceuser admin1password adminadmin1	
		5001 sample_jndi_resource resource executed successfully	
	Where: sample_jndi_resource is the JNDI resource to be deleted.		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-jndi-reso	urce(1), list-jndi-resources(1)	

NAME | delete-jndi-resource – removes the JNDI resource

delete-jvm-options(1)

-)		
delete-jvm-options – delements	leletes the JVM	options from the Java configuration or profiler
[host <i>locall</i> [passwordf [interacti	ost] [port ile filename] ve=true]	min_user [password admin_password] 4848] [secure -s] [terse=false] [echo=false] _option_name=jvm_option_value) [:jvm_option_name=jvm
domain.xml file. You the JVM option starts distinguish that JVM	can enter mor with a dash (-) option is an ope	configuration or Profiler elements of the e than one JVM option separated by a colon (:) . If then use two dashes (—) before the operand to erand and not an option. JVM options are used to articular profiler going.
user	authorized do username.	omain application server administrative
password	password to a	administer the domain application server.
host	machine nam	e where the domain application server is running.
port		of the domain application server listening for n requests.
secure		SL/TLS to communicate with the domain erver.
passwordfile	file containing	g the domain application server password.
terse	avoiding hum	any output data must be very concise, typically nan-friendly sentences and favoring d data for consumption by a script. Default is
echo		e will echo the command line statement on the out. Default is false.
interactive	if set to true (oprompted.	default), only the required password options are
profiler		e JVM options is for the profiler. Profiler must option to be true.
jvm_option_name=jvm_	option_value	the left side of the equal sign (=) is the JVM option name. The right side of the equal sign (=) is the jvm_option_value. Additionally, you can use ":" as a delimiter for more than one jvm-option. If the jvm-option contains a ":", use the escape character \ to offset the ":" delimiter.
	delete-jvm-options - delements delete-jvm-option	delete-jvm-options - deletes the JVM elements delete-jvm-optionsuser ad

EXAMPLES | **EXAMPLE 1** Using delete-jvm-options

```
asadmin> delete-jvm-options --user admin --password adminadmin
--host fuyako --port 7070 --profiler=true -- "-DDebug=true":"-Xmx256m":"
-Dcom.sun.aas.imqBin"="\/export\/as7se\/imq\/bin"
JVM options deleted
```

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

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		
SYNOPSIS	delete-lifecycle-moduleuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] module_name		
DESCRIPTION	Removes the lifecycle module. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS		que identifier for the deployed server lifecycle event listener dule.	
EXAMPLES	EXAMPLE 1 Using delet	e-lifecycle-module	
	asadmin> delete-lifecycle-moduleuser adminpassword adminadminhost fuyakoport 7070 customSetup Deleted the Lifecycle module with module name = customSetup		
	Where: customSetu	up is the lifecycle module deleted.	
EXIT STATUS	0 command exect	ated successfully	
	1 error in executi	ng the command	
SEE ALSO	create-lifecycle	e-module(1), list-lifecycle-modules(1)	

NAME

delete-mime – removes the MIME type

SYNOPSIS

delete-mime

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

DESCRIPTION

Removes the MIME types associated with the named server instance. The server determines the MIME type of a requested resource by invoking the type-by-extension directive in the <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

mime_id

internal name for the MIME types listing. It is used in a virtual-server

element to define the MIME types used by the virtual server.

EXAMPLES

EXAMPLE 1 Using delete-mime

asadmin> delete-mime --user admin --password adminadmin --host fuyako --port 7070 --instance serve Deleted Mime with id = sampleMIME

Where: sampleMIME is the name of the MIME deleted.

EXIT STATUS

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

INTERFACE EQUIVALENT SEE ALSO

HTTP Server node, MIME Type Files page

create-mime(1), list-mimes(1)

delete-persistence-resource(1)

•	` '		
NAME	delete-persistence-res	ource – removes the persistence resource	
SYNOPSIS	[password [secure -s	ce-resourceuser admin_user admin_password][host localhost] [port 4848] t] [passwordfile filename] [terse=false] te] [interactive=true] jndi_name	
DESCRIPTION	Removes the persister	nce resource. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	jndi_name JNDI na	nme of the persistence resource.	
EXAMPLES	EXAMPLE 1 Using delete	-persistence-resource	
	asadmin> delete-persistence-resourceuser admin1password adminadmin1host pigeonport 5001 sample_persistence_resource Command delete-persistence-resource executed successfully Where: sample_persistence_resource is the persistence resource to be deleted.		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-persisten	ce-resource(1), list-persistence-resources(1)	

NAME

delete-profiler – deletes the profiler element

SYNOPSIS

delete-profiler --user admin_user [--password admin_password]

[--host localhost] [--port 4848] [--secure |-s]

[--passwordfile filename] [--terse=false] [--echo=false]

[--interactive=true]

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.

This command is supported in remote mode only.

OPTIONS

user	authorized	domain	application	server administrative

username.

--password password to administer the domain application server.

--host machine name where the domain application server is running.

--port port number of the domain application server listening for

administration requests.

--secure if true, uses SSL/TLS to communicate with the domain

application server.

--passwordfile file containing the domain application server password.

--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. Default is

false.

--echo setting to true will echo the command line statement on the

standard output. Default is false.

--interactive if set to true (default), only the required password options are

prompted.

EXAMPLES

EXAMPLE 1 Using delete-profiler

```
asadmin> delete-profiler --user admin --passwordfile passwords.txt --host localhost --port 4848
```

Deleted Profiler

Where: profiler is the deleted profile element.

EXIT STATUS

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

SEE ALSO

create-profiler(1), list-profiler(1)

delete-resource-adapter-config(1)

ictc-resource-adap	oter-comig(r)		
NAME	delete-resource-adapt domain.xml for the	er-config – deletes the configuration information created in connector module	
SYNOPSIS	delete-resource-adapter-configuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] raname		
DESCRIPTION	Deletes the entry in the file with the provided	ne resource-adapter-config element of the domain.xmlraname.	
	This command is sup	ported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	raname the value k	kept in the resource-adapter-name in the domain.xml file.	
EXAMPLES	EXAMPLE 1 Using delet	te-resource-adapter-config	
	asadmin> delete-resource-adapter-configusername admin1password adminadmin myresourceadapter Command delete-resource-adapter-config executed successfully		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	•	
SEE ALSO		adapter-config(1), list-resource-adapter-configs(1)	

NAME	delete-ssl – deletes the	e ssl element from the HTTP listener,IIOP listener, or IIOP service	
SYNOPSIS	<pre>delete-ssluser admin_user[password admin_password] [host localhost] [port 4848][secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] type [http-listener iiop-listener iiop-service] [listener_id]</pre>		
DESCRIPTION		at from the HTTP listener, IIOP listener, or IIOP service. The red if thetype option is iiop-service.	
	This command is sup	ported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
	type	type of service or listener that the SSL is deleted. The type can be: http-listener, iiop-listener, and iiop-service.	
OPERANDS		of the listener or service that the SSL is deleted. This operand is uired if thetype option is iiop-service.	
EXAMPLES	EXAMPLE 1 Using delete-ssl		
	asadmin> delete-ssluser adminpassword adminadminhost fuyakoport 7070type http-listener http-listener-1 Deleted SSL in HTTP Listener		
	Where: SSL is deleted	for http-listener-1.	
EXIT STATUS	0 command execu	ted successfully	

dal	lete-ssl	11	
ue	1616-221		

1 error in executing the command

SEE ALSO create-ssl(1)

SYNOPSIS delete-threadpool --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] threadpool_id DESCRIPTION Removes the threadpool with the named ID. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. password to administer the domain application server. --password machine name where the domain application server is running. --host port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. file containing the domain application server password. --passwordfile --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. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **OPERANDS** threadpool_id an ID for the work queue; for example, thread-pool-1, thread-pool-2, etc. **EXAMPLES EXAMPLE 1** Using delete-threadpool asadmin> delete-threadpool --user admin1 --password adminadmin1 threadpool-1 Command delete-threadpool executed successfully **EXIT STATUS** 0 command executed successfully 1 error in executing the command **SEE ALSO** create-threadpool(1), list-threadpools(1)

delete-threadpool - removes the named threadpool

NAME

delete-virtual-server(1)

	• •			
NAME	delete-virtual-server – deletes the virtual server with the named virtual server ID			
SYNOPSIS	delete-virtual-serveruser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] virtual_server_id			
DESCRIPTION	Deletes the virtual ser supported in remote r	ver with the named virtual server ID. This command is mode only.		
OPTIONS	user authorized domain application server administrative username.			
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	port	port number of the domain application server listening for administration requests.		
	secure	if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile	file containing the domain application server password.		
	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. Default is false.		
	echo	setting to true will echo the command line statement on the standard output. Default is false.		
	interactive	if set to true (default), only the required password options are prompted.		
OPERANDS		lentifies the unique ID for the virtual server to be created. This irtual server ID cannot begin with a number.		
EXAMPLES	EXAMPLE 1 Using delete-virtual-server			
	asadmin> delete-virtual-serveruser admin1password adminadmin1host pigeonport 5001 sample_vs1 Command delete-virtual-server executed successfully			
	Where sample_vs1 is the virtual server deleted.			
EXIT STATUS	0 command execu	ted successfully		
	1 error in executing the command			
SEE ALSO	create-virtual-s	erver(1), list-virtual-servers(1)		

NAME

deploy - deploys the specified component

SYNOPSIS

deploy --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive] [--virtualservers virtual_servers] [--contextroot context_root] [--force=true] [--precompilejsp=false] [--verify=false] [--name component_name] [--upload=true] [--retrieve local_dirpath] [--dbvendorname dbvendorname] [--createtables=true | false | --dropandcreatetables=true | false] [--uniquetablenames=true | false] [--deploymentplan deployment_plan] [--enabled=true] filepath

DESCRIPTION

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

The --createtables and --dropandcreatetables options are booleans and therefore can take the values of true or false. These options are only used during deployment of CMP beans that have not been mapped to a database (i.e., no sun-cmp-mappings.xml descriptor is provided in the module's META-INF directory). They are ignored otherwise.

The --createtables option and --dropandcreatetables option are mutually exclusive; only one should be used. If drop and/or create tables fails, the deployment does not fail; a warning message is provided in the log file.

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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.

deploy(1)

interactive	prompts you for the required options that are not already specified.
virtualservers	comma separated list of virtual server name.
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	if set to true, the syntax and semantics of the deployment descriptor is verified.
name	name of the deployable component.
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.
retrieve	retrieves the client stub JAR file from the server machine to the local directory.
dbvendorname	name of database vendor being used. Default is the corresponding entry in the cmp-resource() element of the sun-ejb-jar.xml file. If not specified the default is SQL92. Thereby the DDL files to create and drop tables will be generated in SQL92 format.
createtables	creates tables at deploy of an application with unmapped CMP beans. Default is the corresponding entry in the cmp-resource element of the sun-ejb-jar.xml file. If not specified, defaults to the entries specified in the deployment descriptors.
dropandcreatetables	drops tables at redeploy of an already deployed application with unmapped CMP beans. If not specified, the tables will be dropped if the drop-tables-at-undeploy entry in the cmp-resource element of the sun-ejb-jar.xml file is set to true. The new tables are created if the create-tables-at-deploy entry in the cmp-resource element of the sun-ejb-jar.xml

	file is set to true. On redeploy the tables created by the previous deploy are dropped before creating the new tables.
uniquetablenames	guarantees unique table names for all the beans and results in a hashcode added to the table names. This is useful if you have an application with case-sensitive bean names.
deploymentplan	takes the deployment plan, which is a JAR containing Sun-specific descriptors, and deploys it. This should be passed along when deploying a pure EAR file. A pure EAR file is an EAR without Sun-specific descriptors.
enabled	if set to true (default), allows user to access the application. If set to false, user will not be able to access the application.

OPERANDS

filepath

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

EXAMPLES

EXAMPLE 1 Deploying a J2EE application

Deploy (install) the J2EE application packaged in the Cart.ear file.

```
asadmin> deploy --user admin --password admin123 --host murph
--port 4848 Cart.ear
Command deploy executed successfully
```

EXAMPLE 2 Deploying a Web application with the default context root

Deploy the Web application in the hello.war file at the hello context root.

```
asadmin> deploy --user admin hello.war
Command deploy executed successfully
```

EXAMPLE 3 Deploying an enterprise bean (EJB component)

Deploy and enterprise bean with container-managed persistence (CMP) and create the database tables used by the bean.

```
asadmin> deploy --user admin --createtables=true EmployeeEJB.jar
Command deploy executed successfully
```

EXAMPLE 4 Deploying a connector module (resource adapter)

Deploy a connector module packaged in a RAR file.

```
asadmin> deploy --user admin jdbcra.rar
Command deploy executed successfully
```

deploy(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command 1

SEE ALSO

undeploy(1), list-components(1)

NAME

deploydir – deploys an exploded format of application archive

--dropandcreatetables=false] dirpath

SYNOPSIS

deploydir --user admin_user[--password admin_password][--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--virtualservers virtual_servers] [--contextroot context_root] [--force=true] [--verify=false] [--precompilejsp=false] [--name component_name] [--uniquetablenames=true | false]

[--dbvendorname dbvendorname] [--createtables=false |

DESCRIPTION

Deploys the exploded format of the application archives present under the directory provided as the command operand. The deployed format, of EAR or WAR applications, reside on the application server and have a directory structure which can be used for deployment. The --force option makes sure the component is forcefully (re)deployed even if the specified component has already been deployed or already exists. Set --force to false for a first deployment. If the application with that name is running, and force is set to false, the command fails.

If the --uniquetablenames, --createtables, and --dropandcreatetables options are not specified, the entries in the deployment descriptors are used.

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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.

deploydir(1)

interactive	prompts you for the required options that are not already specified.
virtualservers	comma separated list of virtual server IDs.
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	if set to true, the syntax and semantics of the deployment descriptor is verified.
name	name of the deployable component.
uniquetablenames	guarantees unique table names for all the beans and results in a hashcode added to the table names. This is useful if you have an pplication with case-sensitive bean names.
dbvendorname	name of the database vendor being used. Default is the corresponding entry in the cmp-resource () element of the sun-ejb-jar.xml file. If not specified, the default is SQL92. Thereby the DDL files to create and drop tables will be generated in SQL92 format.
createtables	creates tables during deployment for applications using unmapped CMP beans. Default is the corresponding entry in the cmp-resource element of the sun-ejb-jar.xml file. If not specified, defaults to the entries in the deployment descriptors.
dropandcreatetables	drops existing tables and creates tables during deployment for application using unmapped CMP beans. If not specified, the tables will be dropped if the drop-tables-at-undeploy entry in the cmp-resource element of the sun-ejb-jar.xml file is set to true. The new tables are created if the create-tables-at-deploy entry in the cmp-resource element of the sun-ejb-jar.xml is set to true. On redeploy the tables created by the previous deploy are dropped before creating the new tables.

deploydir(1)

OPERANDS

dirpath path to the directory containing the exploded format of the deployable

archive.

EXAMPLES

EXAMPLE 1 Using deploydir

asadmin> deploydir --user admin --passwordfile passwords.txt --host localhost --port 4848 --force=true --precompilejsp=true /home/temp/sampleApp Command deploydir executed successfully

Where: the exploded application to be deployed is in the /home/temp/sampleApp directory.

EXIT STATUS

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

SEE ALSO

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

deploytool(1m)

NAME

deploytool – launches the deploytool utility to deploy, package, and edit your J2EE applications

SYNOPSIS

DESCRIPTION

Use the deploytool utility to deploy and package your J2EE applications and components, create and edit J2EE deployment descriptors, and create and edit J2EE Application Server specific deployment descriptors. If the application is not J2EE compliant, an error message is displayed.

Only one session of the deploytool utility can run with a specific user directory. A lock file is created to ensure that only one utility session is running. A message is displayed if a lock file is detected.

OPTIONS

--help

displays the arguments for launching the Assembly Tool.

--userdir

identifies the user directory. The default user directory is .deploytool under your home directory. Only one deploytool session can be running per user directory. A lock file is created under the user directory to ensure that only one session of the deploytool is running. The deploytool utility uses this directory to store configuration information.

On Solaris, the default directory is at ~/.deploytool

--configdir

identifies the configuration directory. The configuration directory is where the aseny conf file is located.

On Solaris the asenv. conf can be found at:

- Bundled installation: /etc/appserver
- Unbundled installation: default is /etc/opt/SUNWappserver7 or user specified
- Evaluation installation: *AS_SERVER_INSTALL*/config. Where *AS_SERVER_INSTALL* is the directory where you have installed the Java 2 Platform, Enterprise Edition 1.4 Application Server Beta 2.

EXAMPLES

EXAMPLE 1 Using deploytool

example% deploytool --userdir /myapplication --config_dir /myconfigdir

Where --userdir specifies the destination directory, and -config_dir identifies the configuration directory.

SEE ALSO

verifier(1M)

SYNOPSIS	disableuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] component_name			
DESCRIPTION		y disables the named component. The component must have component has not been deployed, an error message is returned.		
OPTIONS	user authorized domain application server administrative username.			
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	port	port number of the domain application server listening for administration requests.		
	secure if true, uses SSL/TLS to communicate with the domain application server.			
	passwordfile file containing the domain application server password.			
	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.		
OPERANDS	component_name name of the component to be disabled.			
EXAMPLES	EXAMPLE 1 Using disable			
	asadmin> disable sampleApp Command disable executed successfully			
EXIT STATUS	0 command executed successfully			
	1 error in executin	g the command		
SEE ALSO	deploy(1), deploydir(1), undeploy(1), enable(1)			

 $\pmb{NAME} \ | \ disable - disables \ the \ component$

display-license(1)

spiay-incerise(1)				
NAME	display-license – displays the license information			
SYNOPSIS	display-license [user admin_user] [password admin_password] [host localhost] [port 4848] [passwordfile filename] [secure -s]			
DESCRIPTION	display-license displays the license information. This command can run both locally and remotely.			
OPTIONS	user	administrative user associated for the instance.		
	password	administrative password corresponding to the administrative user.		
	host	host name of the machine hosting the administrative instance.		
	port	administrative port number associated with the administrative host.		
	passwordfile	file containing passwords appropriate for the command (e.g., administrative instance).		
	secure	if true, uses $\ensuremath{SSL/TLS}$ to communicate with the administrative instance.		
EXAMPLES	asadmin> display-license ************************************			
	Eval Sun ONE Application Server 7 Evaluation License Expiration date Tues 11 Sept 11:58:47 PDT 2002 Number of instances per admin server Unlimited Allow remote administration YES ************************************			
	Expiration date Tu Number of instances Allow remote adminis	n ONE Application Server 7 Evaluation License es 11 Sept 11:58:47 PDT 2002 per admin server Unlimited tration YES ************************************		
EXIT STATUS	0 command execu	ted successfully		
3 2 3	1 error in executin	•		
SEE ALSO	install-license(

11711112	chapte chaptes the component				
SYNOPSIS	enableuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s]passwordfile filename [terse=false] [echo=false] [interactive] component_name				
DESCRIPTION	enable command enables 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. This command is supported in remote mode only.				
OPTIONS	user authorized domain application server administrative username.				
	password	password to administer the domain application server.			
	host	machine name where the domain application server is running.			
	port port number of the domain application server listening for administration requests.				
	secure if true, uses SSL/TLS to communicate with the domain application server.				
	passwordfile file containing the domain application server password.				
	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.			
OPERANDS	component_name name of the component to be enabled.				
EXAMPLES	EXAMPLE 1 Using enable				
2.11.21.22.22.2	asadmin> enable sampleApp Command enable executed successfully				
EXIT STATUS	0 command executed successfully				
	1 error in executing the command				
SEE ALSO	deploy(1), deploydir(1), undeploy(1), disable(1)				

 $\pmb{NAME} \mid enable$ – enables the component

export(1)

NAME |

export – marks a variable name for automatic export to the environment of subsequent commands in multimode

SYNOPSIS

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

DESCRIPTION

Marks a variable name for automatic export to the environment of subsequent commands. All subsequent commands use the variable name values as specified; unless you unset them or exit multimode. If only the variable name is specified, the current value of that variable name is displayed. If the export command is used without any arguments, a list of all the exported variables and their values is displayed. Exported shell environment variables set prior to invoking the asadmin utility are imported automatically and set as exported variables within asadmin. Unexported environment variables cannot be read by the asadmin utility.

OPERANDS

name=value

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

EXAMPLES

EXAMPLE 1 Using export to list the environment variables

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

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

EXIT STATUS

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

SEE ALSO

unset(1), multimode(1)

NAME

freeze-transaction-service – freezes the transaction subsystem

SYNOPSIS

freeze-transaction-service --user admin_user

[--password admin_password] [--host localhost]

[--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true]

DESCRIPTION

Freezes the transaction subsystem during which time all the inflight transactions are suspended. Invoke this command before rolling back any inflight transactions. Invoking this command on an already frozen transaction subsystem has no effect.

This command is supported in remote mode only.

OPTIONS

user	authorized	domain	application	server administrative

username.

--password password to administer the domain application server.

--host machine name where the domain application server is running.

port number of the domain application server listening for --port

administration requests.

if true, uses SSL/TLS to communicate with the domain --secure

application server.

--passwordfile file containing the domain application server password.

indicates that any output data must be very concise, typically --terse

avoiding human-friendly sentences and favoring

well-formatted data for consumption by a script. Default is

false.

setting to true will echo the command line statement on the --echo

standard output. Default is false.

if set to true (default), only the required password options are --interactive

prompted.

EXAMPLES

EXAMPLE 1 Using freeze-transaction-service

asadmin> freeze-transaction-service --user admin --password adminadmin

EXIT STATUS

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

SEE ALSO

unfreeze-transaction-service(1), rollback-transaction(1)

NAME	get – gets the values of the monitorable or configurable attributes				
SYNOPSIS	<pre>getuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] [monitor=false] dotted_attribute_name</pre>				
DESCRIPTION	Gets the values of attributes. If themonitor option is set to true, the monitorable attributes are returned. If themonitor option is set to false, the configurable attribute values are returned. On Solaris, quotes are needed when executing commands with * as the option value or operand.				
OPTIONS	user authorized domain application server administrative username.				
	password	password to administer the domain application server.			
	host	machine name where the domain application server is running.			
	port	port number of the domain application server listening for administration requests.			
	secure if true, uses SSL/TLS to communicate with the domain application server.				
	passwordfile file containing the domain application server password.				
	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. Default is false.				
	echo setting to true will echo the command line statement on the standard output. Default is false.				
	interactive if set to true (default), only the required password options are prompted.				
	monitor	defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are returned.			
OPERANDS	attributename attribute name in the dotted notation.				
EXAMPLES	EXAMPLE 1 Using get asadmin> getuser adminpassword adminadminhost localhostport 4848 "server.resources.jdbc-resource.jdbc/PointBase.*" server.resources.jdbc-resource.jdbc/PointBase.description= <null> server.resources.jdbc-resource.jdbc/PointBase.enabled=true server.resources.jdbc-resource.jdbc/PointBase.jndi-name=jdbc/PointBase server.resources.jdbc-resource.jdbc/PointBase.object-type=user server.resources.jdbc-resource.jdbc/PointBase.pool-name=PointBasePool</null>				

EXIT STATUS | 0

- command executed successfully
- error in executing the command

SEE ALSO

set(1), list(1)

get-client-stubs(1)

NAME	get-client-stubs – gets the stubs of the client				
SYNOPSIS	get-client-stubsuser admin_user [password admin_password]				
DESCRIPTION	Gets the client stubs JAR file for an AppClient standalone module or an application containing the AppClient module, from the server machine to the local directory. This command is supported in remote mode only.				
OPTIONS	user	authorized domain application server administrative username.			
	password	password to administer the domain application server.			
	host	machine name where the domain application server is running.			
	port	port number of the domain application server listening for administration requests.			
	secure	secure if true, uses SSL/TLS to communicate with the domain application server.			
	passwordfile	asswordfile file containing the domain application server password.			
	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.			
	appname	name of the application.			
OPERANDS	local_directory_path	path to the local directory where the client stub should be stored.			
EXAMPLES	EXAMPLE 1 Using get-client-stubs asadmin> get-client-stubsuser adminpassword adminadminhost fuyakoport 7070appname myapplication.ear /sample/exmple				
EXIT STATUS	0 command execu	ted successfully			
	1 error in executin	g the command			
SEE ALSO	undeploy(1)				

NAME

| help – displays a list of all the commands available in the Command-line interface

SYNOPSIS

asadmin help or asadmin --help | -h | -?

DESCRIPTION

The help command displays a list of all the asadmin commands available in the Command-line interface. Specify the command to display the usage information for that command. To display the manpage of each command, use the syntax: asadmin <code>command_name</code> --help | -h | -? or asadmin help <code>command_name</code>

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

add-resources registers the resource in the XML file specified

create-admin-object adds the administered object with the specified

JNDI name

create-audit-module creates an audit module for the optional plugin

module

create-auth-realm adds the new authorized realm

create-connector-connection-pool adds a connection pool with the specified

connection pool name

create-connector-resource registers the resource with the specified JNDI

name

create-connector-security-map creates or modifies a security map for the

namedconnector connection pool

create-custom-resource registers the custom resource

create-domain creates a domain with the given name

create-file-user creates a new file user

create-http-listener adds a new HTTP listener socket

create-iiop-listener adds the IIOP listener

create-javamail-resource registers the Javamail resource

create-jdbc-connection-pool registers the JDBC connection pool

create-jdbc-resource registers the JDBC resource
create-jms-resource registers the JMS resource
create-jmsdest adds the named destination
create-jndi-resource registers the JNDI resource

create-jvm-options creates the JVM options from the Java

configuration or profiler elements

create-lifecycle-module adds a lifecycle module

create-persistence-resource registers the persistence resource

create-profiler creates the profiler element

create-resource-adapter-config creates the resource adapter Java bean

create-ssl creates the SSL element in the HTTP listener or

IIOP listener

create-threadpool creates the thread pool

create-virtual-server adds the named virtual server

delete-admin-object removes the administered object with the

specified JNDI name

delete-audit-module deletes the audit-module for the optional plugin

module

delete-auth-realm removes the named authorized realm delete-connector-connection-pool removes the specified connection pool

delete-connector-security-map deletes the named security map

delete-connector-resource removes the named resource connector

delete-custom-resourceremoves the custom resourcedelete-domaindeletes the given domaindelete-file-userremoves the named file userdelete-http-listenerremoves the HTTP listenerdelete-iiop-listenerremoves the IIOP listener

delete-javamail-resource removes the Javamail resource delete-jdbc-connection-pool removes the JDBC connection pool

delete-jdbc-resource removes the JDBC resource delete-jms-resource removes the JMS resource

delete-jmsdest destroys the named destination delete-jndi-resource removes the JNDI resource

delete-jvm-options deletes the JVM options from the Java

configuration or profiler elements

delete-lifecycle-module removes the lifecycle module delete-persistence-resource removes the persistence resource

delete-profiler deletes the profiler element

delete-resource-adapter-config deletes the resource adapter Java bean

delete-ssl deletes the ssl element from the HTTP listener or

IIOP listener

delete-threadpool deletes the thread pool

delete-virtual-server deletes the virtual server with the named virtual

server ID

deploys the specified component

deploydir deploys the component that is in the directory

located on domain application server

disable stops the component enable runs the component

export marks a variable name for automatic export to

the environment of subsequent commands in

multimode

freeze-transaction-service immobilizes the named transaction service

get gets the values of the monitorable or

configurable attributes

get-client-stubs gets the stubs of the client

help displays a list of all the commands available in

the Command-line interface

jms-ping checks to see if the JMS provider is up and

running

list-admin-objects gets all the administered objects

list-audit-modules lists the audit modules list-auth-realms lists the authorized realms list-components Lists deployed components list-connector-connection-pools gets all the connection pools list-connector-resources gets all the connector resources

list-connector-security-maps lists the security maps for the connector

connection pool

list-custom-resources gets all the custom resources

lists the domains in the given domains directory

list-file-groups lists the file groups list-file-users lists the file users

list-http-listeners gets the HTTP listeners list-iiop-listeners gets the IIOP listeners

list-javamail-resources gets all the Javamail resources

list-jdbc-connection-pools registers the JDBC connection pool

list-jdbc-resources gets all the JDBC resources list-jms-resources gets all the JMS resources

list-jmsdest gets all the named destinations list-jndi-resources gets all the JNDI resources list-lifecycle-modules gets the lifecycle modules

list-persistence-resources gets all the persistence resources

list-resource-adapter-configs lists the resource adapters configured in an

instance

list-sub-components Lists EJBs or Servlets in a deployed module or in

a module of a deployed application

list-threadpools lists the thread pools list-virtual-servers gets the virtual servers

lists the configurable elements

multimode allows you to execute multiple commands while

returning environment settings and remaining in

the asadmin utility

ping-connection-pool tests if a connection pool is usable

rollback-transaction reinitialize the transaction service to its previous

state

set sets the values of attributes

show-component-status displays the status of the deployed component

start-domain starts the given domain stop-domain stops the given domain

undeploy removes a component in the domain application

serve

unfreeze-transaction-service mobilizes the named transaction service unset removes one or more variables from the

multimode environment

update-file-user updates a current file user as specified update-connector-security-map updates the security map for the specified

connector connection pool

verify-domain-xml verifies the content of the domain.xml

version displays the version information

 $\textbf{EXAMPLES} \hspace{0.1cm} | \hspace{0.1cm} \textbf{EXAMPLE 1} \hspace{0.1cm} \textbf{Using the help command}$

asadmin> help

asadmin> create-domain --help

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

SEE ALSO version(1)

install-license(1)

NAME | install-license – installs the license file

SYNOPSIS install-license

DESCRIPTION install-license prevents unauthorized use of the Sun ONE Application Server.

Allows you to install the license file. This command can be run locally only.

EXAMPLES | **EXAMPLE 1** Using install-license

asadmin> install-license

LICENSE agreement will be displayed.

Do you agree with the terms of this license [YES \mid 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	<pre>jms-pinguser admin_user [password admin_password] [host localhost] [port 4848] [passwordfile filename] [secure -s] [terse=false] [echo=false] [interactive=false]</pre>		
DESCRIPTION	Checks to see if the JMS provider is up and running. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	passwordfile file containing the domain application server password.		
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using jms-ping asadmin> jms-pinguser adminpassword adminadminhost bluestarport 4848 server1 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	<pre>jspc [options] jsp_files or jspc [options] -webapp dir</pre>		
DESCRIPTION	Use the jspc command to compile your JSP 1.2 compliant source files into servlets. To allow the application server to pick up the precompiled JSPs from a JAR file, you must disable dynamic reloading of JSPs. To do this, set reload-interval property to -1 in the jsp-config element of the sun-web.xml file.		
	For more information a Developer's Guide.	bout the sun-web.xml file, see the Sun ONE Application Server	
OPTIONS	jsp_files	one or more JSP files to be compiled.	
	-webapp <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 asadmin deploy.	
	-d	enables quiet mode (same as -v0). Only fatal error messages are displayed.	
	-ā dir	the output directory for the compiled JSPs. Package directories are automatically generated based on the directories containing the uncompiled JSPs. The default top-level directory is the directory from which <code>jspc</code> is invoked.	
	-р пате	the name of the target package for all specified JSPs, overriding the default package generation performed by the -d option.	
	-с пате	the target class name of the first JSP compiled. Subsequent JSPs are unaffected.	
	-uribase dir	the URI directory to which compilations are relative. Applies only to JSP files listed in the command, and not to JSP files specified with -webapp option. This is the location of each JSP file relative to the uriroot. If this cannot be determined, the default is /.	
	-uriroot dir	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 -webapp option. If this option is not specified, all parent directories of the first JSP page are searched for a WEB-INF subdirectory. The closest directory to the JSP page that has one is used. If none of the JSP's parent directories have a WEB-INF subdirectory, the directory from which jspc is invoked is used.	
	-genclass	compiles the generated servlets into class files.	

jspc(1M)

-v [level]	enables verbose mode. The level is optional; the default is 2. Possible level values are:
	 0 - fatal error messages only 1 - error messages only 2 - error and warning messages only 3 - error, warning, and informational messages 4 - error, warning, informational, and debugging messages
-mapped	generates separate write calls for each HTML line and comments that describe the location of each line in the JSP file. By default, all adjacent write calls are combined and no location comments are generated.
-die [code]	causes the JVM to exit and generates an error return code if a fatal error occurs. If the code is absent or unparsable it defaults to 1.
-webinc file	creates partial servlet mappings for the -webapp option, which can be pasted into a web.xml file.
-webxml file	creates an entire web.xml file for the -webapp option.
-ieplugin class_id	specifies the Java plugin COM class ID for Internet Explorer. Used by the jsp:plugin tags.

EXAMPLES

EXAMPLE 1 Using jspc to compile the JSPs in a web application

The following command compiles a set of JSP files into Java files under Hellodir:

jspc -d Hellodir welcome.jsp shop.jsp checkout.jsp

The following command compiles all the JSP files in the specified webapp into class files under Hellodir:

jspc -d Hellodir -genclass -webapp /path_to_webapp_directory

To use these precompiled JSP in the web application, put the generated files under Hellodir into a JAR file, place the JAR file under WEB-INF/lib and set reload-interval property to -1 in the jsp-config element of the WEB-INF/sun-web.xml file.

SEE ALSO

asadmin(1M)

NAME | list – lists the configurable elements

SYNOPSIS

list --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--monitor=false] dotted_parent_element_name

DESCRIPTION

Lists the configurable element. On Solaris, quotes are needed when executing commands with * as the option value or operand.

The dotted notation follows these guidelines:

- Any list command that has a dotted name that is not followed by a wildcard (*) will get, as its result, the current node's immediate children. For example, list --monitor server lists all immediate children belonging to the server node.
- Any list command that has a dotted name followed by a wildcard(*) will get, as its result, a hierarchical tree of children nodes from the current node. For example, list --monitor server.applications.* will list all children of applications and their subsequent child nodes and so on.
- Any list command that has a dotted name preceded or followed by a wildcard (*) of the form *dotted name or dotted * name or dotted name* will get, as its result, all nodes and their children matching the regular expression created by the provided matching pattern.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
monitor	defaults to false; if set to false, the configurable attribute values are returned. If set to true, the monitorable attribute values are

returned.

OPERANDS

dotted_parent_element_name configurable or monitorable element name.

EXAMPLES

EXAMPLE 1 Using list to view all dotted-name prefixes

```
asadmin> list --user admin --password adminadmin
--port 5001 "*"
server
server.admin-service
server.admin-service.das-config
server.application-ref.MEjbApp
server.application-ref.__ejb_container_timer_app
server.application-ref.adminapp
server.application-ref.admingui
server.application-ref.com_sun_web_ui
server.applications
server.applications.j2ee-application.MEjbApp
server.applications.j2ee-application.__ejb_container_timer_app
server.applications.web-module.adminapp
server.applications.web-module.admingui
server.applications.web-module.com sun web ui
server.ejb-container
server.http-service
server.http-service.http-listener.admin-listener
server.http-service.http-listener.http-listener-1
server.http-service.http-listener.http-listener-2
server.iiop-service
server.iiop-service.iiop-listener.SSL
server.iiop-service.iiop-listener.SSL.ssl
server.iiop-service.iiop-listener.SSL_MUTUALAUTH
server.iiop-service.iiop-listener.SSL MUTUALAUTH.ssl
server.iiop-service.iiop-listener.orb-listener-1
server.iiop-service.orb
server.java-config
server.jms-service
server.jms-service.jms-host.default JMS host
server.log-service
server.log-service.module-log-levels
server.mdb-container
server.monitoring-service
\verb|server.monitoring-service.module-monitoring-levels|\\
server.resource-ref.jdbc/PointBase
server.resource-ref.jdbc/ TimerPool
server.resources
server.resources.jdbc-connection-pool.PointBasePool
\verb|server.resources.jdbc-connection-pool.\__TimerPool|
server.resources.jdbc-resource.jdbc/PointBase
server.resources.jdbc-resource.jdbc/ TimerPool
server.security-service
server.security-service.audit-module.default
server.security-service.auth-realm.certificate
server.security-service.auth-realm.file
server.security-service.jacc-provider.default
server.thread-pools
server.thread-pools.thread-pool.thread-pool-1
server.transaction-service
```

EXIT STATUS

SEE ALSO

```
EXAMPLE 1 Using list to view all dotted-name prefixes
                                                      (Continued)
server.virtual-server.__asadmin
server.virtual-server.server
server.web-container
EXAMPLE 2 Using list for an application
asadmin> list --user admin --password adminadmin
--host localhost --port 4848 server.applications.j2ee-application
server.applications.j2ee-application.MEjbApp
server.applications.j2ee-application._ejb_container_timer_app
server.applications.j2ee-application.stateless-simple
EXAMPLE 3 Using list for a web module
asadmin> list --user admin --password adminadmin
--host localhost --port 4848 server.applications.web-module
server.applications.web-module.adminapp
server.applications.web-module.adminguip
server.applications.web-module.com sun web ui
0
     command executed successfully
1
     error in executing the command
get(1), set(1)
```

NAME | list-acls – gets the access control lists

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

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

OPTIONS --user administrative user associated for the instance.

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

--secure indicates communication with the administrative instance in

secured mode.

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

administrative instance).

OPERANDS *instance_name* name of the instance.

EXAMPLES | **EXAMPLE 1** Using list-acls

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

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

EXIT STATUS 0 command executed successfully

1 error in executing the command

INTERFACE EQUIVALENT SEE ALSO Access Control List page

create-acl(1), delete-acl(1)

list-admin-objects(1)

NAME	list-admin-objects – go	ets all the administered objects		
SYNOPSIS	list-admin-objectsuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]			
DESCRIPTION	Lists all the administe	ered objects. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.		
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	port	port number of the domain application server listening for administration requests.		
	secure	if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile file containing the domain application server password.			
	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. Default is false.		
	echo	setting to true will echo the command line statement on the standard output. Default is false.		
	interactive	if set to true (default), only the required password options are prompted.		
EXAMPLES	EXAMPLE 1 Using list-ad	lmin-objects		
	asadmin> list-admin-	objectsuser adminpassword admin123 bjects executed successfully		
EXIT STATUS	0 command execu	ted successfully		
	1 error in executin	•		
SEE ALSO		ect(1), delete-admin-object(1)		

list-audit-modules – lists all the audit-modules NAME **SYNOPSIS** list-audit-modules --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] **DESCRIPTION** Lists all the audit modules. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **EXAMPLES EXAMPLE 1** Using list-audit-modules asadmin> list-audit-modules --user admin1 --password adminadmin1 --host pigeon --port 5001 Command list-audit-modules executed successfully **EXIT STATUS** 0 command executed successfully error in executing the command 1 **SEE ALSO** create-audit-module(1), delete-audit-module(1)

list-authdbs(1)

NAME | list-authdbs – gets the authorized database **SYNOPSIS** list-authdbs --user admin_user[--password admin_password][--host localhost] [--port 4848] [--passwordfile filename] [--secure | -s] [--instance instance_name] --virtualserver virtualserver_ID authdb_ID **DESCRIPTION** Gets the access control lists associated with the named server instance. **OPTIONS** --user administrative user associated for the instance. administrative password corresponding to the administrative --password --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). if true, uses SSL/TLS to communicate with the administrative --secure instance. virtual server ID. It can also be referred to as the variable \$id --virtualserver in an obj.conf file. A virtual server ID cannot begin with a number. name of the instance. **OPERANDS** instance_name **EXAMPLES EXAMPLE 1** Using list-authdbs asadmin> lsit-authdbs --user admin --password adminadmin --host fuyako --port 7070 --virtualserver default sampleAuth Where: default and sampleAuth are the authdb IDs in virtual server server1 and instance server1 listed. asadmin% list-authdbs --instance server1 **EXIT STATUS** 0 command executed successfully error in executing the command

SEE ALSO

create-authdb(1), delete-authdb(1)

list-auth-realms – lists the authorized realms NAME **SYNOPSIS** list-auth-realms --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] **DESCRIPTION** Lists the authorized realms. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **EXAMPLES EXAMPLE 1** Using list-auth-realms asadmin> list-auth-realms --user admin --password adminadmin --host localhost --port 4848 file ldap certificate Command list-auth-realms executed successfully Where file, ldap, certificate, and db are the auth realms listed. **EXIT STATUS** 0 command executed successfully error in executing the command **SEE ALSO** create-auth-realm(1), delete-auth-realm(1)

list-components(1)				
NAME	list-components – Lists deployed components			
SYNOPSIS	list-componentsuser admin_user[password admin_password][host localhost] [port 4848][secure -s][passwordfile filename] [terse=false] [echo=false] [interactive] [type application ejb web connector]			
DESCRIPTION	list-components lists your deployed J2EE components. If thetype option is not specified, all components are listed. The available type values are: application (default), ejb, web, and connector. This command is supported in remote mode only.			
OPTIONS	user authorized domain application server administrative username.			
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	 port port number of the domain application server listening fo administration requests. secure if true, uses SSL/TLS to communicate with the domain application server. 			
	passwordfile	file containing the domain application server password.		
	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.		
	type	identifies the type of component to be listed; defaults to application.		
EXAMPLES	EXAMPLE 1 Using list-	components		
	asadmin> list-componentstype application sampleApp J2EE-application Command list-components executed successfully			
	Where: the applications that were deployed are listed.			
EXIT STATUS	0 command execut	ted successfully		
	1 error in executing	g the command		

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

SYNOPSIS list-connector-connection-pools --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] **DESCRIPTION** gets all the connector connection pools. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. password to administer the domain application server. --password machine name where the domain application server is running. --host port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. file containing the domain application server password. --passwordfile --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. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. if set to true (default), only the required password options are --interactive prompted. **EXAMPLES EXAMPLE 1** Using list-connector-connection-pools asadmin> list-connector-connection-pools --user admin --password adminadmin **EXIT STATUS** 0 command executed successfully 1 error in executing the command **SEE ALSO** create-connector-connection-pool(1), delete-connector-connection-pool(1)

list-connector-connection-pools – gets all the connection pools

NAME

list-connector-resources(1)

NAME	list-connector-resourc	es – gets all the connector resources
SYNOPSIS	[password [secure -s	resourcesuser admin_user admin_password] [host localhost] [port 4848] s] [passwordfile filename] [terse=false] se] [interactive=true]
DESCRIPTION	Gets all the connector	resources. This command is supported in remote mode only.
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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. Default is false.
	echo	setting to true will echo the command line statement on the standard output. Default is false.
	interactive	if set to true (default), only the required password options are prompted.
EXAMPLES	EXAMPLE 1 Using the lis	st-connector-resources command
	asadmin> list-connec	tor-resourcesuser admin1 n1host pigeonport 5001
	This will list all the co	onnector resources that have been created.
EXIT STATUS	0 command execu	ted successfully
	1 error in executin	ng the command
SEE ALSO	create-connector	-resource(1), delete-connector-resource(1)

NAME

list-connector-security-maps – lists the security maps for the named connector connection pool

SYNOPSIS

list-connector-security-maps --user admin_user

[--password admin_password] [--host localhost] [--port 4848] [--secure|-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--verbose=false] [--securitymap mapname] pool_name

DESCRIPTION

lists the security map belonging to the named connector connection pool.

This command is supported in remote mode only.

OPTIONS

user	authorized domain	application	server administrative
------	-------------------	-------------	-----------------------

username.

--password password to administer the domain application server.

--host machine name where the domain application server is running.

--port port number of the domain application server listening for

administration requests.

--secure if true, uses SSL/TLS to communicate with the domain

application server.

--passwordfile file containing the domain application server password.

--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. Default is

false.

--echo setting to true will echo the command line statement on the

standard output. Default is false.

--interactive if set to true (default), only the required password options are

prompted.

--verbose lists the identify, principals, and the security name.

--securitymap name of the security map.

OPERANDS

poolname name of the pool.

EXAMPLES

EXAMPLE 1 Using list-connector-security-maps with security map option

asadmin> list-connector-security-maps --user admin --password adminadmin --securitymap mysecuremap securityPool1

Command list-connector-security-maps executed successfully

One security map (mysecuremap) is listed for the securityPool1 pool.

list-connector-security-maps(1)

EXAMPLE 2 Using list-connector-security-maps

 $\verb| asadmin> list-connector-security-maps -- user admin -- password adminadmin securityPool1 \\$

Command list-connector-security-maps executed successfully

All the security maps are listed for the securityPool1 pool.

EXIT STATUS

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

SEE ALSO

 $\label{lem:delete-connector-security-map} \ensuremath{(1)}, \ensuremath{\texttt{create-connector-security-map}(1)}, \ensuremath{\texttt{update-connector-security-map}(1)}$

NAME list-custom-resources – gets all the custom resources **SYNOPSIS** list-custom-resources --user admin_user[--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] **DESCRIPTION** Gets all the custom resources. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **EXAMPLES EXAMPLE 1** Using list-custom-resources asadmin> list-custom-resources --user admin1 --password adminadmin1 --host pigeon --port 5001 sample custom resource Where: sample custom resource is the custom resource listed. **EXIT STATUS** 0 command executed successfully 1 error in executing the command

create-custom-resource(1), delete-custom-resource(1)

SEE ALSO

list-domains(1)

st-domains(1)			
NAME	list-domains – lists the domains in the given domains directory		
SYNOPSIS	list-domains [domaindir install_dir/domains] [terse=false] [echo=false]		
DESCRIPTION	list-domains lists the domains in the given domains directory. This command is supported in local mode only.		
OPTIONS	domaindir	directory where the domains are located. If specified, path must be accessible in the filesystem. If not specified, the domains in the default <code>install_dir/domains</code> directory are listed.	
	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.	
EXAMPLES	EXAMPLE 1 Using list-domains		
	asadmin> list-domains List of domains: domain1 running samples not running Where: the domain1 and samples are listed and their status printed.		
EXIT STATUS	0 command executed successfully		
	1 error in exec	uting the command	
SEE ALSO	create-domain	(1), delete-domain(1), start-domain(1), stop-domain(1)	

NAME | list-file-groups – lists the file groups

SYNOPSIS

list-file-groups --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s]

[--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] [--authrealmname auth_realm_name]

[--name username]

DESCRIPTION

Lists the available groups in the file user. If the name option is not specified, then all groups are listed.

This command is supported in remote mode only.

OPTIONS

user	authorized	domain	application	server administrative
ubci	authorized	aomani	application	server administrative

username.

password to administer the domain application server. --password

--host machine name where the domain application server is

running.

port number of the domain application server listening for --port

administration requests.

--secure if true, uses SSL/TLS to communicate with the domain

application server.

file containing the domain application server password. --passwordfile

indicates that any output data must be very concise, typically --terse

avoiding human-friendly sentences and favoring

well-formatted data for consumption by a script. Default is

false.

setting to true will echo the command line statement on the --echo

standard output. Default is false.

if set to true (default), only the required password options are --interactive

prompted.

--authrealmname filename where you have different stores for file auth realm.

name of the file user. --name

EXAMPLES

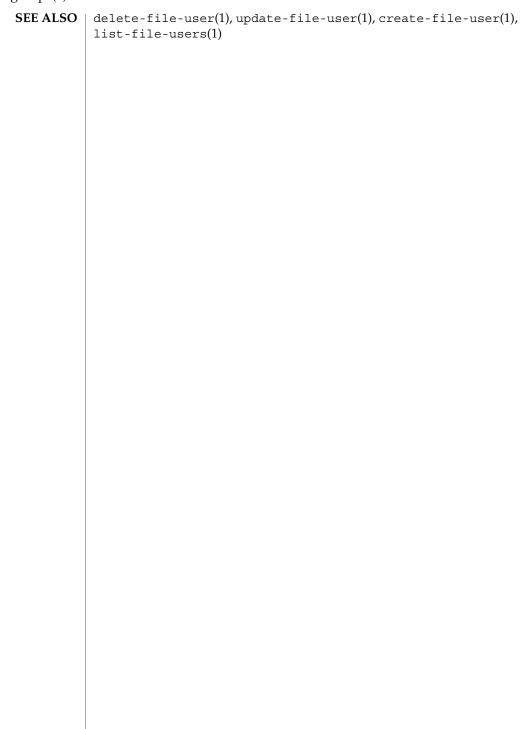
EXAMPLE 1 Using the list-file-groups command

```
asadmin> list-file-groups --user admin1 --password adminadmin1
--host pigeon --port 5001 --name sample_user
Command list-file-groups executed successfully
```

EXIT STATUS

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

list-file-groups(1)



NAME

list-file-users – lists the file users

SYNOPSIS

list-file-users --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false]

[--interactive=true] [--authrealmname auth_realm_name]

DESCRIPTION

Lists all the file users. If the *auth_realm_name* is not specified, an entry is created in the default keyfile. If auth_realm_name is specified, an entry is created in the keyfile where

OPTIONS

the auth-realm name	in the domain.xml file points to. This command is supported
in remote mode only.	
user	authorized domain application server administrative

--password password to administer the domain application server. --host

machine name where the domain application server is

running.

username.

port number of the domain application server listening for --port

administration requests.

if true, uses SSL/TLS to communicate with the domain --secure

application server.

--passwordfile file containing the domain application server password.

indicates that any output data must be very concise, typically --terse

avoiding human-friendly sentences and favoring

well-formatted data for consumption by a script. Default is

setting to true will echo the command line statement on the --echo

standard output. Default is false.

if set to true (default), only the required password options are --interactive

prompted.

--authrealmname name, in the domain.xml file, where you have different

stores for file auth realm.

EXAMPLES

EXAMPLE 1 Using the list-file-users command

```
asadmin> list-file-users --user admin1 --password adminadmin1
--host pigeon --port 5001 --authrealmname auth-realm1
Command list-file-user executed successfully
```

EXIT STATUS

- 0 command executed successfully
- error in executing the command

SEE ALSO

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

list-http-listeners(1)

NAME	list-http-listeners – gets the HTTP listeners		
SYNOPSIS	[host localh	ersuser admin_user [password admin_password] ost] [port 4848] [secure -s] tile filename] [terse=false] [echo=false] ve=true]	
DESCRIPTION	Gets the HTTP listeners. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using list-ht	tp-listeners	
	asadmin> list-http-1:host pigeonport http-listener-1 http-listener-2 admin-listener	istenersuser admin1password adminadmin1	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-http-list	<pre>ener(1), delete-http-listener(1)</pre>	

NAME	list-iiop-listeners – ge	ts the IIOP listeners	
SYNOPSIS	list-iiop-listenersuser admin_user[password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Gets the IIOP listeners. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using list-iid	pp-listeners	
	asadmin> list-iiop-listenersuser adminpassword adminadminhost fuyakoport 7070 orb-listener-1 sample_iiop_listener Where: orb-listener-1 and sample_iiop_listener are the IIOP listeners listed.		
EXIT STATUS	0 command execu	ited successfully	
	1 error in executin		
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 admin_user] [--password admin_password] [--host localhost] [--port 4848] [--domain domain_name] [--local=false] [--passwordfile filename] [--secure | -s]

DESCRIPTION

Use the list-instances to list all the instance in the server. The list-instances

OPTIONS

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.			
	a durinistrative uses associated for the instances		
user	administrative user associated for the instanace.		

password	administrative	password corres	ponding t	to the administrative

user.

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

port number associated with the administrative host. --port

--domain name of the domain.

--local determines if the command should delegate the request to

administrative instance or run locally.

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

administrative instance).

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

instance.

EXAMPLES

EXAMPLE 1 Using list-instances in local mode

```
asadmin> list-instances --domain1 --local
admin-server running
server1 running
```

Where: the server1 and admin-server instances for the domain1 domain is listed.

EXAMPLE 2 Using list-instances in remote mode

```
asadmin> list-instances --user admin --passwordfile passwords.txt --host localhost --port 4848
server1 [mayank:80] running
```

Where: the server1 instance associated with the specified user, passwords, host, and port number specified is listed for the remote machine.

EXIT STATUS

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

INTERFACE EQUIVALENT

Server Instance page

SEE ALSO | show-instance-status(1)

NAME	list-javamail-resources – gets all the Javamail resources		
SYNOPSIS	[host localh	sourcesuser admin_user [password admin_password] ost] [port 4848] [secure -s] tile filename] [terse=false] [echo=false] ve=true]	
DESCRIPTION	Gets all the Javamail 1	resources. This command can only be run remotely.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
EXAMPLES	EXAMPLE 1 Using list-jav	vamail-resources	
	asadmin> list-javamail-resourcesuser admin1password adminadmin1host pigeonport 5001 mail/MyMailSession		
	Command list-javamai	l-resources executed successfuly	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	•	
SEE ALSO		resource(1), delete-javamail-resource(1)	

list-jdbc-connection-pools(1)

t jabe connection	poois(1)		
NAME	list-jdbc-connection-pools – gets all the JDBC connection pools		
SYNOPSIS	list-jdbc-connection-poolsuser admin_user [password admin_password] [host localhost] [port 4848] [[secure -s]passwordfile filename] [terse=false] [echo=false] [interactive]		
DESCRIPTION	Gets all the JDBC con	nection pools. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
EXAMPLES	EXAMPLE 1 Using list-jdl	bc-connection-pools	
	asadmin> list-jdbc-co	onnection-poolsuser admin nhost fuyakoport 7070 server	
	Where: XA_connectio	n_pool is the JDBC connection listed.	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-jdbc-conn	ection-pool(1), delete-jdbc-connection-pool(1)	
	I		

SYNOPSIS	list-jdbc-resourcesuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Lists all the JDBC reso	ources. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
EXAMPLES	EXAMPLE 1 Using the lis	t-jdbc-resources command	
	asadmin> list-jdbc-resourcesuser admin1password adminadmin1host pigeonport 5001 sample_jdbc_resource Command list-jdbc-resources executed successfully		
	Where: sample_jdbo	c_resource is the JDBC resource listed.	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-jdbc-reso	urce(1), delete-jdbc-resource(1)	

 $\begin{tabular}{ll} NAME & | list-jdbc-resources - gets all the JDBC resources \\ \end{tabular}$

list-jmsdest(1)

NAME	list-jmsdest – gets all the physical destinations		
SYNOPSIS	list-jmsdestuser admin_user[password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] [desttype -T topic queue]		
DESCRIPTION	Gets all the physical JMS destinations. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
	desttype	type of JMS destination. Valid values are topic, and queue.	
EXAMPLES	EXAMPLE 1 Using list-	jmsdest	
	asadmin> list-jmsdestuser adminpassword adminadminhost bluestarport 4848 server PhysicalQueue queue {} PhysicalTopic topic {} Command list-jmsdest executed successfully		
EXIT STATUS	0 command execut	ted successfully	
	1 error in executin	g the command	
SEE ALSO	create-jmsdest(1),	delete-jmsdest(1)	

NAME | list-jmsobj – gets all the named objects

SYNOPSIS list-jmsobj --instance instance_name --objtype | -o object_type

DESCRIPTION Gets all the named destination objects.

OPTIONS --instance the name of the instance.

--objtype

EXAMPLES asadmin% list-jmsobj --instance server1 --objtype xxxx

INTERFACE EQUIVALENT

unknown

SEE ALSO create-jmsobj(1) delete-jmsobj(1)

list-jms-resources(1)

NAME	list-jms-resources – gets all the JMS resources		
SYNOPSIS	[password [secure -s	esuser admin_user admin_password] [host localhost] [port 4848]] [passwordfile filename] [terse=false] e] [interactive=true] [restype resource_type]	
DESCRIPTION	Gets all the JMS resources. This command can only be run remotely.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
	restype	JMS resource type which can be: javax.jms.Topic, javax.jms.Queue, javax.jms.TopicConnectionFactory, javax.jms.QueueConnectionFactory.	
EXAMPLES	EXAMPLE 1 Using the lis	t-jms-resources command to list all JMS resources	
	asadmin> list-jms-resources jms/Queue jms/Topic jms/QueueConnectionFactory jms/DurableTopicConnectionFactory Command list-jms-resources executed successfully		
	EXAMPLE 2 Using the list-jms-resources command to list JMS resources of a specified type		
	asadmin> list-jms-resourcesrestype javax.jms.TopicConnectionFactory jms/DurableTopicConnectionFactory jms/TopicConnectionFactory		
	Command list-jms-resources executed successfully		

list-jms-resources(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command

SEE ALSO

create-jms-resource(1), delete-jms-resource(1)

list-jndi-resources(1)

NAME	list-jndi-resources – gets all the JNDI resources		
SYNOPSIS	list-jndi-resourcesuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Gets all the JNDI reso	ources. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
EXAMPLES	EXAMPLE 1 Using the list-jndi-resource command		
	asadmin> list-jndi-resourceuser admin1password adminadmin1host pigeonport 5001 Command list-jndi-resources executed successfully		
EXIT STATUS	0 command executed successfully		
	1 error in executing the command		
SEE ALSO			

list-lifecycle-modules – gets the lifecycle modules NAME **SYNOPSIS** list-lifecycle-modules --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] DESCRIPTION Gets the lifecycle modules. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **EXAMPLES EXAMPLE 1** Using list-lifecycle-modules asadmin> list-lifecycle-modules --user admin --password adminadmin --host fuyako --port 7070 customSetup Where: customSetup is the lifecycle module listed. **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

SYNOPSIS

list-mimes

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

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 <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.

OPERANDS

instance_name name of the instance.

EXAMPLES

EXAMPLE 1 Using list-mimes

asadmin> list-mimes --user admin --password adminadmin --host fuyako --port 7070 server1 sampleMIME

Where: sampleMIME is the name of the MIME listed.

EXIT STATUS

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

INTERFACE EQUIVALENT SEE ALSO

HTTP Server node, MIME Type Files page

create-mime(1), delete-mime(1)

SYNOPSIS list-persistence-resources --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure | -s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] DESCRIPTION Gets all the persistence resources. This command is supported in remote mode only. **OPTIONS** --user authorized domain application server administrative username. --password password to administer the domain application server. --host machine name where the domain application server is running. port number of the domain application server listening for --port administration requests. --secure if true, uses SSL/TLS to communicate with the domain application server. --passwordfile file containing the domain application server password. indicates that any output data must be very concise, typically --terse avoiding human-friendly sentences and favoring well-formatted data for consumption by a script. Default is false. --echo setting to true will echo the command line statement on the standard output. Default is false. --interactive if set to true (default), only the required password options are prompted. **EXAMPLES EXAMPLE 1** using list-persistence-resources asadmin> list-persistence-resources --user admin1 --password adminadmin1 --host pigeon --port 5001 Command list-persistence-resources executed successfully Where: sample persistence resource is the persistence resource listed. **EXIT STATUS** 0 command executed successfully 1 error in executing the command **SEE ALSO** create-persistence-resource(1), delete-persistence-resource(1)

list-persistence-resources – gets all the persistence resources

NAME

list-profiler(1)

NAME | list-profiler – gets the profiler element in the named instance.

SYNOPSIS list-profiler

--user user_name --password password --host hostname --port admin_port_number

[--instance instance_name]

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

INTERFACE Application Server Instances, JVM Settings tab **EQUIVALENT SEE ALSO**

create-profiler(1) delete-profiler(1)

NAME |

list-profilers – lists the profiler elements

SYNOPSIS

list-profilers --user admin_user[--password admin_password]

[--host localhost]

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

DESCRIPTION

Gets the profiler element associated with the named server instance.

OPTIONS

--user administrative user associated for the instance.

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

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

administrative instance).

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

instance.

OPERANDS

instance_name name of the instance.

EXAMPLES

EXAMPLE 1 Using list-profilers

asadmin> list-profilers --user admin --passwordfile passwords.txt --host localhost --port 4848 ser

sample_profiler

Where: sample profiler is the profiler listed.

EXIT STATUS

0 command executed successfully

1 error in executing the command

INTERFACE EQUIVALENT SEE ALSO Application Server Instances, JVM Settings tab

create-profiler(1), delete-profiler(1)

list-resource-adapter-configs(1)

NAME	list-resource-adapter-configs – lists the configuration information created in domain.xml for the connector module			
SYNOPSIS	<pre>list-resource-adapter-configsuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] [verbose=false] [ranameconnector_module_name]</pre>			
DESCRIPTION	_	n information in the domain.xml for the connector module. It esource-adapter-config in the domain.xml.		
	This command is supp	ported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.		
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	port	-port port number of the domain application server listening for administration requests.		
	secure	-secure if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile	file containing the domain application server password.		
	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. Default is false.			
	echo setting to true will echo the command line statement on the standard output. Default is false.			
	interactive	if set to true (default), only the required password options are prompted.		
	verbose	if set to true the properties that are configured are also listed. Default is false.		
OPERANDS	raname the value kept in the resource-adapter-name in the domain.xml file.			
EXAMPLES	EXAMPLE 1 Using list-resource-adapter-configs			
	asadmin> list-resource-adapter-configsusername admin1			
	password adminadmin1 Command list-resource-adapter-configs executed successfully			
EXIT STATUS	0 command execu	ted successfully		
	1 error in executing the command			

$\textbf{SEE ALSO} \ | \ \texttt{create-resource-adapter-config(1)},$

delete-resource-adapter-config(1)

list-sub-components(1)

NAME	list-sub-components – Lists EJBs or Servlets in a deployed module or in a module of a deployed application		
SYNOPSIS	list-sub-componentsuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] [appname app_name] module_name		
DESCRIPTION	module of the deploye	nts lists your EJBs or Servlets in a deployed module or in a ed application. If a module is not identified, all modules are is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile	file containing the domain application server password.	
	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.	
	appname	name of the application.	
OPERANDS	module_name name	e of the module containing the sub-component.	
EXAMPLES	EXAMPLE 1 Using list-sub-components		
	asadmin> list-sub-componentsappname sampleApp mymodule sampleApp application		
EXIT STATUS	0 command evecu	tod successfully	
LAIT STATES	0 command executed successfully 1 error in executing the command		
SEE ALSO	<pre>deploy(1), deploydir(1), undeploy(1), enable(1), disable(1), list-components(1)</pre>		

		list-threadpools(1)	
NAME	list-threadpools – lists all the threadpools		
SYNOPSIS	list-threadpoolsuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Lists all the thread po	ols. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile file containing the domain application server password.		
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using list-threadpools		
		poolsuser adminpassword adminadmin pools executed successfully	
EXIT STATUS	0 command executed successfully		
	1 error in executin	•	
SEE ALSO	create-threadpool(1), delete-threadpool(1)		

list-virtual-servers(1)

NAME	list-virtual-servers – gets the virtual servers		
SYNOPSIS	list-virtual-serversuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Gets all the virtual ser	rver elements. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.		
	echo setting to true will echo the command line statement on the standard output. Default is false.		
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using list-virtual-servers		
	asadmin> list-virtual-serversuser adminpassword adminadminhost localhostport 4848 Command list-virtual-servers executed successfully		
EXIT STATUS	0		
EXIT STATUS	0 command executed successfully		
SEE ALGO	1 error in executing the command		
SEE ALSO	create-virtual-server(1), delete-virtual-server(1)		

NAME

multimode – allows you to execute multiple commands while preserving environment settings and remaining in the asadmin utility

SYNOPSIS

DESCRIPTION

Use multimode to process the asadmin commands. The command-line interface will prompt you for a command, execute that command, display the results of the command, and then prompt you for the next command. Additionally, all the asadmin option names set in this mode are used for all the subsequent commands. You can set your environment and run commands until you exit multimode by typing "exit" or "quit." You can also provide commands by passing a previously prepared list of commands from a file or standard input (pipe). You can invoke multimode from within a multimode session; once you exit the second multimode environment, you return to your original multimode environment.

This command is supported in local mode only.

OPTIONS

--file reads the commands as defined in the file.

--printprompt allows the printing of asadmin prompt after each command is

executed. Set this option to false when the commands are piped or redirected from the standard input or file. By default the

option is set to true.

--encoding specifies the locale for the file to be decoded.

--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. Default is false.

--echo setting to true will echo the command line statement on to the

standard output. Default is false.

EXAMPLES

EXAMPLE 1 Using multimode to execute multiple commands

example% asadmin multimode --file commands file.txt

Where: example% is the system prompt. The multimode settings are executed from the commands file.txt file.

EXIT STATUS

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

SEE ALSO

export(1), unset(1)

package-appclient(1M)

NAME

package-appclient – packs the application client container libraries and jar files

SYNOPSIS

package-appclient

DESCRIPTION

Use the package-appclient command to pack the application client container libraries and jar files into an appclient.jar file. The created file is located at <code>appserver_install_dir/lib/appclient/appclient.jar</code>. The appclient.jar file provides an application client container package targeted at remote hosts that do not contain a server installation.

The appclient.jar 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 appclient.jar produced on a Solaris SPARC platform cannot be used on a Windows client machine.

After copying the appclient.jar file to a remote location, unjar it to get a set of libraries and jar files in the appclient directory

After unjarring on the client machine, modify appclient_install_dir/config/asenv.conf (asenv.bat for Windows) as follows:

- set AS WEBSERVICES LIB to appclient_install_dir/lib
- set AS NSS to appclient_install_dir/lib (appclient_install_dir\bin for Windows)
- set AS IMQ LIB to appclient_install_dir/imq/lib
- set AS INSTALL to appclient_install_dir
- set AS JAVA to your JDK 1.4 home directory
- set AS ACC CONFIG to appclient_install_dir/config/sun-acc.xml

Modify appclient_install_dir/config/sun-acc.xml as follows:

- Ensure the DOCTYPE file references *appclient install dir*/lib/dtds
- Ensure that target-server address attribute refrences the server machine.
- Ensure that target-server port attribute refrences the ORB port on the remote machine.
- Ensure that log-service references a log file; if the user wants to put log messages to a log file.

Modify appclient_install_dir/bin/appclient (appclient.bat for Windows) as follows:

■ change token %CONFIG HOME% to appclient_install_dir/config

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE		
Interface Stability	Unstable		

SEE ALSO | appclient (1M)

ping-connection-pool(1)

N	۸	N	E
IN	А	IVI	Е

ping-connection-pool – tests that a connection pool is usable

SYNOPSIS

ping-connection-pool --user admin_user [--password admin_password] [--host localhost] [--port 4848] [--secure |-s] [--passwordfile filename] [--terse=false] [--echo=false] [--interactive=true] pool_name

DESCRIPTION

Tests that a connection pool is usable for both JDBC connection pools and connector connection pools. For example, if you create a new JDBC connection pool for use with an application that is expected to be deployed, before deploying the application, the previously created pool is tested with this command.

This command is supported in remote mode only.

OPTIONS

user	authorized domain application server administrative username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is

--echo

setting to true will echo the command line statement on the standard output. Default is false.

--interactive

if set to true (default), only the required password options are prompted.

OPERANDS

poolname name of the connection pool to be tested.

EXAMPLES

EXAMPLE 1 Using ping-connection-pool

Before you can ping a connection pool, you must:

- Create a connection pool with authentication.
- Make sure the Enterprise Information Server (can be the database) is started.

```
asadmin> ping-connection-pool --user admin1 --password adminadmin1
--host pigeon --port 5001 sampleConnectionPool
Command ping-connection-pool executed successfully
```

ping-connection-pool(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command

SEE ALSO

```
create-connector-connection-pool(1),
delete-connector-connection-pool(1),
\verb|list-connection-pool| (1), \verb|create-jdbc-connection-pool| (1), \\
delete-jdbc-connection-pool(1)list-jdbc-connection-pools(1)
```

reconfig(1)

NAME

reconfig – applies the changes you have made for a server instance

SYNOPSIS

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

DESCRIPTION

This command is deprecated. It allows you to apply changes you have made for a server instance. Use the reconfig command after you've used the set command to change server properties. Any changes you make to the configuration files of the server do not take affect until you apply the changes by running the reconfig command. When --discardmanualchanges is set to true, manual changes made to the server.xml file are discarded. When --keepmanualchanges is set to true, manual changes made to the server.xml file take affect. However if both options are false (both options are not specified), an error message is displayed if manual changes and/or changes have been applied using the Administrator Interface. Use this command with discretion since there is no undo, and the changes applied are made directly to your config/backup/server.xmlfile.

OPTIONS

user	administrative user associated for the instance.

--password administrative password corresponding to the

administrative user.

--host host name of the machine hosting the administrative

instance.

--port administrative port number associated with the

administrative host.

--passwordfile file containing passwords appropriate for the

command (e.g., administrative instance).

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

administrative instance.

--discardmanualchanges defaults to false. When set to true, discards the

changes made manually to the server.xml file.

--keepmanualchanges defaults to false. When set to true, allows the

manual changes made to the server.xml file to

take affect.

OPERANDS

instance_name name of the instance..

EXAMPLES

EXAMPLE 1 Using reconfig

asadmin> reconfig --user admin

--passwordfile passwords.txt --host localhost --port 4848 server1

Successfully reconfigured

reconfig(1)

EXAMPLE 2 Using reconfig with the --discardmanual changes option

asadmin> reconfig --user admin --passwordfile passwords.txt --host localhost --port 4848 --discard Instance restart is required Successfully reconfigured

EXAMPLE 3 Using reconfig with the --keepmanual changes option

asadmin> reconfig --user admin --passwordfile passwords.txt --host localhost --port 4848 --keepman Instance restart is required Successfully reconfigured

EXIT STATUS

- 0 command executed successfully
- error in executing the command

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

restart-instance(1)

NAME

restart-instance – restarts the specified server instance and all the services associated with it

SYNOPSIS

restart-instance [--user admin_user] [--password admin_password]

[--host localhost] [--port 4848] [--local=false]
[--domain domain_name] [--passwordfile filename]

[--secure|-s] instance_name

DESCRIPTION

Use the restart-instance to restart the instance with the instance name specified. The restart-instance command can be run both locally and remotely. To restart remotely, the administration server must be running on the hostname and port number specified. The user authenticates using the password identified for the administration server. Additionally, the instance must already exist within the domain served by the administration server, and the instance must be running. The restart-instance command is not supported on Windows.

OPTIONS

	1	• • 1 6 • 1 • •
user	administrative user a	associated for the instance.

password	administrative	password corres	ponding to	the administrative

user.

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

--port administrative port number associated with the administrative

host.

--local determines if the command should delegate the request to

administrative instance or run locally.

--domain name of the domain.

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

administrative instance).

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

instance.

OPERANDS

instance_name name of the instance to be restarted.

EXAMPLES

EXAMPLE 1 Using restart-instance in local mode

asadmin> restart-instance --local --domain domain1 server1 Instance server1 started

Where: server1 is the name of the instance restarted on the domain1 domain.

 $\textbf{EXAMPLE 2} \ Using \ \texttt{restart-instance} \ in \ remote \ mode$

asadmin> restart-instance --user admin --password adminadmin --host bluestar --port 4848 server1 Instance server1 started

Where: server1 is the name of the instance restarted. The restarted instance is associated with the user, password, host, and port number specified.

restart-instance(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command

INTERFACE EQUIVALENT SEE ALSO

Server Instance page

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

rollback-transaction(1)

iiback-transaction	(1)		
NAME	rollback-transaction – rollsback the named transaction		
SYNOPSIS	unfreeze-transaction-serviceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] transaction_id		
DESCRIPTION	Rollsback the named	transaction. This command is supported in remote mode only.	
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile file containing the domain application server password.		
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	transaction_id ider	ntifier for the transaction to be rolled back	
EXAMPLES	EXAMPLE 1 Using rollback-transaction		
	asadmin> rollback-transactionuser adminpassword adminadmin transaction1		
EXIT STATUS	0 command executed successfully		
	1 error in executing the command		
SEE ALSO	freeze-transaction-service(1), unfreeze-transaction-service(1)		
l l			

SYNOPSIS	<pre>setuser admin_user[password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] attributename=value [attribute_name=value] *</pre>		
DESCRIPTION	Sets the values of one or more configurable attribute. This command is supported in remote mode only. On Solaris, quotes are needed when executing commands with * as the option value or operand.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
OPERANDS	attributename=value	identifies the attribute name and its value. See the <i>Application Server Reference</i> for a listing of the available attribute names.	
EXAMPLES	EXAMPLE 1 Using set		
	asadmin> setuser adminpassword adminadminhost localhostport 4848 server.transaction-service.automatic-recovery=true		
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	•	
SEE ALSO	get(1), list(1)		

 $\pmb{NAME} \ | \ set-sets \ the \ values \ of \ attributes$

show-component-status(1)

NAME	show-component-status – displays the status of the deployed component		
SYNOPSIS	show-component-statususer admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] component_name		
DESCRIPTION	show-component-status gets the status of the deployed component. The status is a string representation returned by the server. The possible status strings include: enabled or disabled. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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.	
OPERANDS	component_name n	ame of the component to be listed.	
EXAMPLES	EXAMPLE 1 Using show-component-status		
	asadmin> show-component-status sampleApplication		
	Status of sampleApplication is enabled		
	Where: the status of the	he sampleApplication application is shown.	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	list-components(1	1), list-sub-components(1)	

NAME | shutdown – brings down the administration server

SYNOPSIS

shutdown

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

DESCRIPTION

shutdown gracefully brings down the administration server and all the running instances. You must manually start the administration server to bring it up again.

OPTIONS

--user administrative user associated for the instance.

--password administrative password corresponding to the administrative

user

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

--port administrative port number associated with the administrative

host.

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

administrative instance).

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

instance.

EXAMPLES

EXAMPLE 1 Using the shutdown command

asadmin> shutdown --user admin --password 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

INTERFACE EQUIVALENT SEE ALSO

Administration Server page

```
start-instance(1), stop-instance(1),
restart-instance(1)start-domain(1), stop-domain(1), start-appserv(1),
stop-appserv(1)
```

start-appserv(1)

NAME

start-appserv – starts the domains in the specified or default domains directory

SYNOPSIS

start-appserv [--domaindir install_dir/domains]
[--terse=false] [--echo=false]

DESCRIPTION

This command has been deprecated. Please use start-domain to start the domains in the specified domains directory. If a domain directory is not specified, the default domains directory is started. If there is only one domain in the default domain directory (<code>install_dir/domains</code>), then no operand is required to start the domain. However, if there is more than one domain, the domain operand must be specified using the <code>start-domain</code> command. This command can only be run locally.

OPTIONS

--domaindir directory where the domains are started. If specified, path must be

accessible in the filesystem. If not specified, the domain is started in

the default <code>install_dir/domains</code> directory.

--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. Default is false.

--echo setting to true will echo the command line statement on the

standard output. Default is false.

EXIT STATUS

0 command executed successfully

1 error in executing the command

SEE ALSO

 $\label{eq:create-domain} $$\operatorname{create-domain}(1), \operatorname{stop-domain}(1), \operatorname{delete-domain}(1), \\ \operatorname{list-domains}(1)$$

NAME	start-domain – starts the given domain		
SYNOPSIS	start-domain [domaindir install_dir/domains] [terse=false] [echo=false] [verbose=false] [debug] [domain_name]		
DESCRIPTION	start-domain starts the specified domain. If there is only one domain in the default domain directory (<i>install_dir</i> /domains), then no operand is required to start the domain. However, if there is more than one domain, the domain operand must be specified. This command can only be run locally.		
OPTIONS	domaindir	directory where the domain is to be started. If specified, path must be accessible in the filesystem. If not specified, the domain is started in the default <code>install_dir/domains</code> directory.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	verbose	by default this flag is set to false. If set to true, detailed server startup output is displayed. Press CTRL-C to kill the server. Press CTRL-\\ to print a thread dump.	
	debug	by default this flag is set to false. If set to true, the server is started in debug mode and prints the JPDA port on the console.	
OPERANDS	domain_name name of the domain. Must be a unique name.		
EXAMPLES	EXAMPLE 1 Using start-domain		
	asadmin> start-domain sampleDomain Starting Domain sampleDomain, please wait. Domain sampleDomain started		
	Where: the sampleDomain domain is started.		
EXIT STATUS	0 command ex	xecuted successfully	
	1 error in executing the command		

SEE ALSO create-domain(1), delete-domain(1), stop-domain(1), list-domains(1)

start-instance(1)

NAME

start-instance - starts a server instance and all the services associated with it

SYNOPSIS

start-instance [--user admin_user] [--password admin_password]

[--host local_host] [--port 4848] [--local=false]

[--domain domain_name] [--debug=false] [--passwordfile filename]

[--secure|-s] instance_name

DESCRIPTION

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

OPTIONS

	1	16 (1
user	administrative user	associated for the instance.

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

--local determines if the command should delegate the request to

administrative instance or run locally.

--domain name of the domain.

--debug starts the instance in debug mode.

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

administrative instance).

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

instance.

OPERANDS

instance_name name of the instance to be started.

EXAMPLES

EXAMPLE 1 Using start-instance in local mode

asadmin> start-instance --domain domain1 admin-server

Instance admin-server started

Where: the admin-server instance is started on the local domain1 domain.

EXAMPLE 2 Using start-instance in remote mode

asadmin> start-instance --user admin --password bluestar --host localhost --port 4848 server1 Instance server1 started

Where: the server1 instance is started on the remote domain associated with the specified user, password, host, and port number.

start-instance(1)

EXIT STATUS | 0

- command executed successfully
- error in executing the command 1

INTERFACE EQUIVALENT SEE ALSO

Server Instance page

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

stop-appserv(1)

NAME

stop-appserv – stops the domains in the specified or default domains directory

SYNOPSIS

stop-appserv [--domaindir install_dir/domains]
 [--terse=false] [--echo=false]

DESCRIPTION

This command has been deprecated. Please use stop-domain to stop the domains in the specified or default domains directory. If the domains directory is not specified, the default domains directory is stopped. If there is only one domain in the default domains directory (<code>install_dir/domains</code>), then no operand is required to stop the domain. However, if there is more than one domain, the domain operand must be specified using the <code>stop-domain</code> command. This command is supported in local mode only.

OPTIONS

--domaindir directory where the domains are started. If specified, path must be

accessible in the filesystem. If not specified, the domain is started in

the default *install_dir/* domains directory.

--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. Default is false.

--echo setting to true will echo the command line statement on the

standard output. Default is false.

EXIT STATUS

command executed successfully

1 error in executing the command

SEE ALSO

 $\verb|create-domain|(1), \verb|start-domain|(1), \verb|stop-domain|(1), \verb|delete-domain|(1), \\ \verb|list-domain|(1)|$

NAME

stop-domain – stops the given domain

SYNOPSIS

stop-domain [--domaindir install_dir/domains] [--terse=false]
 [--echo=false] [domain_name]

DESCRIPTION

stop-domain stops the specified domain. If there is only one domain in the default domain directory (*install_dir*/domains), then no operand is required to stop the domain. However, if there is more than one domain, the domain operand must be specified. This command is supported in local mode only.

OPTIONS

specified. This command is supported in local mode only.

--domaindir directory where the domain is to be stopped. If specified, path

must be accessible in the filesystem. If not specified, the domainis stopped in the default <code>install_dir/domains</code> directory.

--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. Default is false.

--echo setting to true will echo the command line statement on the

name of the domain. Must be a unique name.

standard output. Default is false.

OPERANDS

domain_name

EXAMPLES | EXAMPL

EXAMPLE 1 Using stop-domain asadmin> stop-domain sampleDomain

Stoping Domain sampleDomain, please wait

Domain sampleDomain stopped

Command stop-domain executed successfully

Where: the sampleDomain domain in the default domains directory is stopped.

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)

stop-instance(1)

TA T		78.	Æ	
1	Δ	10.	/	н

stop-instance – stops the specified server instance and all the services associated with it

SYNOPSIS

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

DESCRIPTION

Use the stop-instance to stop the instance with the instance name specified. The stop-instance can be run both locally and remotely. The named instance must already exist within the given domain; and the instance must be running.

OPTIONS

already exist within the	ne given domain; and the instance must be running.
user	administrative user associated for the instance.

--password administrative password corresponding to the administrative

user.

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

--port administrative port number associated with the administrative

host.

--local determines if the command should delegate the request to

administrative instance or run locally.

--domain name of the domain.

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

administrative instance).

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

instance.

OPERANDS

instance_name name of the instance to be stopped.

EXAMPLES

EXAMPLE 1 Using stop-instance in local mode

asadmin> stop-instance --local --domain domain1 server1 Instance server1 stopped

Where: the server1 instance associated with the domain1 domain is stopped locally.

EXAMPLE 2 Using stop-instance in remote mode

asadmin> stop-instance --user admin --password bluestar --host localhost --port 4848 server1 Instance server1 stopped

Where: the server1 instance associated with the named user, password, host and port is deleted from the remote machine.

EXIT STATUS

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

INTERFACE EQUIVALENT

Server Instance page

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)

I - 7 (-)				
NAME	undeploy – removes a component in the domain application server			
SYNOPSIS	undeployuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [-interactiv=truee] [droptables=true false] [cascade=false] component_name			
DESCRIPTION	undeploy removes the specified component in the domain application server.			
	Thedroptables option is only used to undeploy CMP beans for which the tables had been created by the deployment. If not specified, the entries in the deployment descriptors are used.			
	This command is supported in remote mode only.			
OPTIONS	user	authorized domain application server administrative username.		
	password	password to administer the domain application server.		
	host	machine name where the domain application server is running.		
	port	port number of the domain application server listening for administration requests.		
	secure	if true, uses SSL/TLS to communicate with the domain application server.		
	passwordfile	file containing the domain application server password.		
	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.		
	droptables	if set to true, tables created by application using CMP beans during deployment are dropped. Default is the corresponding entry in the cmp-resource element of the sun-ejb-jar.xml file. If not specified, defaults to the entries specified in the deployment descriptors.		
	cascade	If set to true, it deletes all the connection pools and connector resources associated with the resource adapter being undeployed. If set to false, the undeploy fails if any pools and resources are still associated with the resource adapter. Then, either those pools and resources have to be deleted explicitly, or the option has to be set to true. If the option is set to false, and if there are no pools and resources still associated with the		

resource adapter, the resource adapter is undeployed. This option is applicable to connectors (resource adapters) and applications.

OPERANDS

component_name

name of the deployable component.

EXAMPLES

EXAMPLE 1 Simple undeployment

Undeploy (uninstall) an application named Cart

asadmin> undeploy --user admin Cart

EXAMPLE 2 Undeploying an enterprise bean with container-managed persistence (CMP)

Undeploy a CMP bean named myejb and drop the corresponding database tables. In a production environment, database tables contain valuable information, so use the --droptables option with care.

asadmin> undeploy --user admin --droptables=true myejb

EXAMPLE 3 Undeploy a connector (resource adapter)

Undeploy the connector module named jdbcra and perform a cascading delete to remove the associated resources and connection pools.

asadmin> undeploy --user admin --cascade=true jdbcra

EXIT STATUS

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

SEE ALSO

deploy(1), deploydir(1), list-components(1)

unfreeze-transaction-service (1)

NAME	unfreeze-transaction-s	service – resumes all suspended transactions	
SYNOPSIS	unfreeze-transaction-serviceuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true]		
DESCRIPTION	Resumes all the suspended inflight transactions. Invoke this command on an already frozen transaction. This command is supported in remote mode only.		
OPTIONS	user	authorized domain application server administrative username.	
	password	password to administer the domain application server.	
	host	machine name where the domain application server is running.	
	port	port number of the domain application server listening for administration requests.	
	secure	if true, uses SSL/TLS to communicate with the domain application server.	
	passwordfile	file containing the domain application server password.	
	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. Default is false.	
	echo	setting to true will echo the command line statement on the standard output. Default is false.	
	interactive	if set to true (default), only the required password options are prompted.	
EXAMPLES	EXAMPLE 1 Using unfree	eze-transaction-service	
		ansaction-serviceuser adminpassword adminadmin	
EXIT STATUS	0 command execu	ted successfully	
	1 error in executin	g the command	
SEE ALSO	freeze-transacti	on-service(1), rollback-transaction(1)	

unset – removes one or more variables from the multimode environment NAME

SYNOPSIS unset env_var [env_var] *

DESCRIPTION

Removes one or more variables you set for the multimode environment. The variables and their associated values will no longer exist in the environment.

OPERANDS

env_var environment variable to be removed.

EXAMPLES

EXAMPLE 1 Using unset to remove environment variables

asadmin> export AS ADMIN HOST=bluestar AS ADMIN PORT=8000 AS ADMIN USER=admin AS ADMIN PASSWORD=pa asadmin> export AS ADMIN PREFIX=server1.jms-service asadmin> export AS ADMIN HOST=bluestar AS ADMIN PORT=8000 AS_ADMIN_USER=admin AS ADMIN PASSWORD=***** AS_ADMIN_PREFIX=server1.jms-service asadmin> unset AS ADMIN PREFIX asadmin> export AS ADMIN HOST=bluestar AS ADMIN PORT=8000 AS_ADMIN_USER=admin AS ADMIN PASSWORD=******

Using the export command without the argument lists the environment variables that are set. Notice the AS ADMIN PREFIX is not in the environment after running the unset command.

EXIT STATUS

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

SEE ALSO export(1), multimode(1)

update-connector-security-map(1)

NAME	update-connector-security-map – updates the security map for the named connector connection pool	
SYNOPSIS	update-connector-security-mapuser admin_user [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive=true] poolname connector_connection_pool_name [addprincipals principal-name[, principal-name]*] [addusergroups user-group[, user-group]*] [removeprincipals principal-name[, principal-name]*] [removeusergroups user-group[, user-group]*] [mappedusername user_name] [[mappedpassword password]] mapname	
DESCRIPTION	Modifies a security map for the named connector connection pool. You must have first created a connector connection pool using the create-connector-connection-pool command.	
	This command is supported	d in remote mode only.
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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. Default is false.
	echo	setting to true will echo the command line statement on the standard output. Default is false.
	interactive	if set to true (default), only the required password options are prompted.
	poolname	connector connection pool name for which the security map that is to be updated or created belongs to.
	addprincipals	a comma separated list of backend EIS principals to be added.

update-connector-security-map(1)

--addusergroups a comma separated list of the enterprise information

system usergroups to be added.

--removeprincipals a comma separated list of the enterprise information

system principals to be removed.

--removeusergroups a comma separated list of the enterprise information

system usergroups to be removed.

--mappedusername the enterprise information system username.--mappedpassword the enterprise information system password.

OPERANDS

mapname name of the security map to be updated.

EXAMPLES

EXAMPLE 1 Using update-connector-security-map

It is assumed that the connector pool has already been created using the create-connector-pool command.

asadmin> update-connector-security-map --user admin --password adminadmin --poolname connector-pool1 --addprincipals principal1, principal2,

--addusergroups usergroup1, usergroup2 --removeprincipals principal3, principal4

--removeusergroups usergroup3, usergroup4 securityMap1

Command update-connector-security-map executed successfully

EXIT STATUS

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

SEE ALSO

create-connector-security-map(1), delete-connector-security-map(1),
list-connector-security-maps(1)

update-file-user(1)

N	A	Λ	41	

update-file-user - updates a current file user as specified

SYNOPSIS

update-file-user --user admin_user [--password admin_password]
 [--host localhost] [--port 4848] [--secure|-s]
 [--passwordfile filename] [--terse=false] [--echo=false]
 [--interactive=true] [--userpassword user_password]
 [--groups user_groups:[user_groups]*] [--authrealmname
 auth_realm_name] user_name

DESCRIPTION

Updates an existing entry in keyfile by the specified user_name, user_password and groups. Multiple groups can be entered by separating them, with a colon, ":". If the <code>auth_realm_name</code> is not specified, an entry is created in the default keyfile. If <code>auth_realm_name</code> is specified, an entry is created in the keyfile where the <code>auth-realm_name</code> name in the <code>domain.xml</code> file points to.

authorized domain application server administrative

This command is supported in remote mode only.

OPTIONS

--user

4501	username.
password	password to administer the domain application server.
host	machine name where the domain application server is running.
port	port number of the domain application server listening for administration requests.
secure	if true, uses SSL/TLS to communicate with the domain application server.
passwordfile	file containing the domain application server password.
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. Default is false.
echo	setting to true will echo the command line statement on the standard output. Default is false.
interactive	if set to true (default), only the required password options are prompted.
userpassword	password for the file user.
groups	group where the file user belongs to.
authrealmname	name, in the domain.xml file, where you have different stores for file auth realm.
user name of file user	

OPERANDS | *user_name*

user_name name of file user.

 $\textbf{EXAMPLES} \hspace{0.1cm} | \hspace{0.1cm} \textbf{EXAMPLE 1} \hspace{0.1cm} \textbf{Using the update-file-user command}$

```
asadmin> update-file-user --user admin1 --password adminadmin1
--host pigeon --port 5001 --userpassword sample_password
--groups staff:manager:engineer --authrealmname auth-realm1 sample_user
Command update-file-user executed successfully
```

Where: the sample_user is the file user updated with the updated user password and groups.

EXIT STATUS

- 0 command executed successfully
- error in executing the command

SEE ALSO

```
delete-file-user(1), list-file-users(1), create-file-user(1),
list-file-groups(1)
```

verifier(1M)

NAME

verifier – validates the J2EE Deployment Descriptors against application server DTDs

SYNOPSIS

verifier [-v] [-d destination_directory] [-r [a|w|f]] jar_filename

DESCRIPTION

Use the verifier utility to validate the J2EE deployment descriptors and the Sun ONE Application Server specific deployment descriptors. If the application is not J2EE compliant, an error message is printed.

When you run the verifier utility, two results files are created in XML and TXT format. The location where the files are created can be configured using the -d option. The directory specified as the destination directory for result files should exist. If no directory is specified, the result files are created in the current directory. Result files are named as jar_filename verified.xml and jar_filename verified.txt

The XML file has various sections that are dynamically generated depending on what kind of application or module is being verified. The root tag is staticverification which may contain the tags application, ejb, web, appclient, connector, other, error and failure-count. The tags are self explanatory and are present depending on the type of module being verified. For example, an EAR file containing a web and EJB module will contain the tags application, ejb, web, other, and failure-count.

If the verifier ran successfully, a result code of 0 is returned. A non-zero error code is returned if the verifier failed to run.

OPTIONS

- verbose debugging is turned on.
- -d identifies where the result files get placed.
- identifies the reporting level defined as one of the following: -r
 - a sets output reporting level to display all results (default)
 - w sets output reporting level to display warning and failure results
 - f sets output reporting level to display only failure results

jar_filename

name of the ear/war/jar file to perform static verification on. The results of verification are placed in two files jar_filename verified.xml and jar_filename verified.txt in the destination directory.

EXAMPLES

EXAMPLE 1 Using verifier in the Verbose Mode

example% verifier -v -d /verifier-results -rf sample.ear

Where -v runs the verifier in verbose mode, -d specifies the destination directory, and -rf displays only the failures. The results are stored in /verifierresults/sample.ear verified.xml and /verifierresults/sample.ear verified.txt.

SEE ALSO

asadmin(1M)

verify-domain-xml(1)

NAME | verify-domain-xml – verifies the content of the domain.xml

SYNOPSIS | verify-domain-xml [--verbose=false]

[--domaindir install_dir/domains] domain_name

DESCRIPTION Verfies the content of the domain.xml file.

OPTIONS --verbose turns on (true) or off (false, default) verbose debugging.

--domaindir directory where the domain is to be created. If specified, path must

be accessible in the filesystem. If not specified, the domain is

created in the default <code>install_dir/domains</code> directory.

OPERANDS | *domain_name* name of the domain. Must be a unique name.

EXAMPLES | **EXAMPLE 1** Using the verify-domain-xml command

asadmin> verify-domain-xml --verbose=true domain1

Element: applications

Error: J2eeApplication Module does not contains application name 'MEjbApp'

J2eeApplication Module does not contains application name '__ejb_container_timer_app'

EXIT STATUS 0 command executed successfully

1 error in executing the command

version(1)

ersion(1)		
NAME	version – displays the	version information
SYNOPSIS	<pre>version [user admin_user] [password admin_password] [host localhost] [port 4848] [secure -s] [passwordfile filename] [terse=false] [echo=false] [interactive] [verbose=false]</pre>	
DESCRIPTION	version displays the version information. If the command-line cannot communicate with the administration server with the given user/password and host/port, then the command-line will retrieve the Version locally and display a warning message. If the user option is not entered, the command-line will retrieve the version locally and display a warning message. The warning message will not be displayed ifterse option is entered on the command line.	
OPTIONS	user	authorized domain application server administrative username.
	password	password to administer the domain application server.
	host	machine name where the domain application server is running.
	port	port number of the domain application server listening for administration requests.
	secure	if true, uses SSL/TLS to communicate with the domain application server.
	passwordfile	file containing the domain application server password.
	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	EXAMPLE 1 Using remot	e mode to display version
	asadmin> version	
	Java 2 Platform Enterprise Edition 1.4 Application Server	
	EXAMPLE 2 Using remote mode to display version in detail	
	asadmin> versionuser adminpassword adminadmin	
	host bluestarport 4848verbose Java 2 Platform Enterprise Edition 1.4 Application Server (build A021930-126949)	

EXIT STATUS 0 command executed successfully

error in executing the command

SEE ALSO help(1)

wscompile(1M)

NAME

wscompile – generates stubs, ties, serializers, and WSDL files used in JAX-RPC clients and services

SYNOPSIS

wscompile [options] configuration_file

DESCRIPTION

Generates the client stubs and server-side ties for the service definition interface that represents the web service interface. Additionally, it generates the WSDL description of the web service interface which is then used to generate the implementation artifacts.

In addition to supporting the generation of stubs, ties, server configuration, and WSDL documents from a set of RMI interfaces, wscompile also supports generating stubs, ties and remote interfaces from a WSDL document.

You must specify one of the -gen options in order to use wscompile as a stand alone generator. You must use either -import (for WSDL) or -define (for an RMI interface) along with the -model option in order to use wscompile in conjunction with wsdeploy.

Invoking the wscompile command without specifying any arguments outputs the usage information.

OPTIONS

-cp *path* location of the input class files.

-classpath path same as -cp path option.

-d *directory* where to place the generated output files.

-define read the service's RMI interface, define a service. Use this

option with the -model option in order to create a model

file for use with the wsdeploy command.

-f: features enables the given features. Features are specified as a

comma separated list of features. See the list of supported

features below.

-features: features same as -f: features option.

-g generates the debugging information.

-gen generates the client-side artifacts.-gen:both generates client and server artifacts.

-gen:client same as -gen option.

-qen:server generates the server-side artifacts and the WSDL file. If

you are using wsdeploy, you do not specify this option.

-httpproxy: host:port specifies an HTTP proxy server; defaults to port 8080.

-import reads a WSDL file, generates the service RMI interface and

a template of the class that implements the interface. Use

this option with the -model option in order to create a

model file for use with the wsdeploy command.

-model write the internal model for the given file name. Use this

option with the -import option in order to create a model file for use with the wsdeploy command.

-keep keeps the generated files.

-nd *directory* directory for the non-class generated WSDL files are

stored.

-0 optimizes the generated code.

-s *directory* directory for the generated source files.

-verbose output messages about what the compiler is doing.

-version prints version information.

Exactly one of the -input, -define, -gen options must be specified.

SUPPORTED FEATURES

datahandleronly always map attachments to data handler type

documentliteral use document literal encoding

donotoverride do not regenerate classes that already exist in the classpath.

donotunwrap disable unwrapping of document/literal wrapper elements in

WSI mode (default).

explicitcontext turn on explicit service context mapping.

infix:name specify an infix to use for generated serializers (Solaris). infix=name specify an infix to use for generated serializers (Windows).

jaxbenumtype map anonymous enumeration to its base type.

nodatabinding turn off data binding for literal encoding.

noencodedtypes turn off encoding type information.

nomultirefs turn off support for multiple references.

norpcstructures do not generate RPC structures (-import only).

novalidation turn off validation for the imported WSDL file.

resolveidref resolve xsd:IDREF.

rpclietral use the RPC literal encoding.

searchschema search schema aggresively for subtypes.

serializeinterfaces turn on direct serialization of interface types.

strict generate code strictly compliant with JAXRPC specification.

wscompile(1M)

unwrap enable unwrapping of document/literal wrapper elements in

WSI mode.

useonewayoperations allow generation of one-way operations.

wsi enable WSI-Basic Profile features, to be used for

document/literal, and RPC/literal.

Note: the -gen options are not compatible with wsdeploy.

CONFIGURATION FILE

The wscompile command reads the configuration file config.xml which contains information that describes the web service. The structure of the file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration
xmlns="http://java.sun.com/xml/ns/jax-rpc/ri/config">
<service> or <wsdl> or <modelfile>
</configuration>
```

The configuration element may contain exactly one <service>, <wsdl> or <modelfile>.

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"/>

MODELFILE ELEMENT

This element is for advanced users.

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.xml.gz"/>

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.

SEE ALSO

wsdeploy(1M)

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 -o input_WAR_file options

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.

Generally, you don't have to run wsdeploy because the functions it performs are done automatically when you deploy a WAR with deploytool or asadmin.

OPTIONS

-classpath *path* location of the input class files.

-keep keep temporary files.

-tmpdir temporary directory to use.

-o output WAR file required; location of the generated WAR file.

-source *version* generates code for the specified JAX-RPC SI version.

Supported version are: 1.0.1, 1.0.3, and 1.1 (default).

-verbose outputs messages about what the compiler is doing.

-version prints version information.

INPUT WAR FILE

The input WAR file for wsdeploy will typically have the following structure:

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 endpoint interface, and HelloImpl is the class thatimplements the interface. The web.xml file is tghe deployment descriptor of a web component.

jaxrpc-ri.xml FILE

The following is a simple HelloWorld service.

```
<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"</pre>
```

```
displayName="HelloWorld Service"
description="A simple web service"
wsdl="/WEB-INF/<wsdlname>
interface="hello.HelloIF"
implementation="hello.HelloImpl"/>
<endpointMapping
endpointName="MyHello"
urlPattern="/hello"/>
</webServices>
```

The webServices() element must contain one or more endpoint() elements. The interface and implementation attriutes of endpoint() specify the service's interface and implementation class. The endpointMapping() element associates the service port with the part of the endpoint URL path that follows the urlPatternBase().

NAMESPACE MAPPINGS

Here is a schema type name example:

```
schemaType="ns1:SampleType" xmlns:ns1="http://echoservice.org/types"
```

When generating a Java type from a schema type, wscompile gets the classname from the local part of the schema type name. To specify the package name of the generated Java classes, you define a mapping between the schema type namespace and the package name. You define this mapping by adding a <namespaceMappingRegistry> element to the config.xml file. For example:

```
<service>
...
    <namespaceMappingRegistry>
        <namespaceMapping
        namespace="http://echoservice.org/types"
        packageName="echoservice.org.types"/>
        </namespaceMappingRegistry>
.....
</service>
```

You can also map namespaces in the oppisite direction, from schema types to Java types. In this case, the generated schema types are taken from the package that the type comes from.

HANDLERS

A handler accesses a SOAP message that represents an RPC request or response. A handler class must implement the <code>javax.xml.rpc.handler</code> interface. Because it accesses a SOAP message, a handler can manipulate the message with the APIs of the <code>javax.xml.soap.package()</code>.

A handler chain is a list of handlers. You may specify one handler chain for the client and one for the server. On the client, you include the handlerChains() element in the jaxrpc-ri.xml file. On the server, you include this element in the config.xml file. Here is an example of the handlerChains() element in the config.xml:

wsdeploy(1M)

```
<handlerChains>
 <chain runAt="server"</pre>
   roles=
    "http://acme.org/auditing
    "http://acme.org/morphing" xmlns:ns1="http://foo/foo-1">
   <handler className="acme.MyHandler"</pre>
    headers ="ns1:foo ns1:bar"/>
    property
     name="property" value="xyz"/>
    </handler>
   </chain>
</handlerChains>
```

For more information on handlers, see the SOAP message Handlers chapter of the JAX-PRC specifications.

SEE ALSO

wscompile(1M)

Index

a

applies the changes you have made for a server instance — reconfig, 198

b

brings down the administration server and associated instances — shutdown, 205

d

displays the license information — display-license, 138 displays the status of the deployed component — show-component-status, 204

g

generates stubs, ties, serializers, and WSDL files used in JAX-RPC clients and services — wscompile, 226

i installs the license file — install-license, 150

L

Lists deployed components —
list-components, 162
Lists EJBs or Servlets in a deployed module or
in a module of a deployed application —
list-sub-components, 190

r

reads a WAR file and the jaxrpc-ri.xml file and generates another WAR file that is ready for deployment — wsdeploy, 230 registers the persistence resource create-persistence-resource, 79 enables the component — enable, 139

S

sets the values of attributes — set, 203 starts the domains in the specified or default domains directory — start-appserv, 206 disables the component — disable, 137

Α

add-resources — registers the resource in the XML file specified, 14
adds a new access control list file for the named instance — create-acl, 29

- adds a new HTTP listener socket create-http-listener, 51 adds an audit-module — create-auditmodule, 32 adds the IIOP listener — create-iiop-listener, 55 adds the physical destination create-jmsdest, 67 adds the named virtual server create-virtual-server, 89 adds the new authorized database for the named instance — create-authdb, 34 adds the new authorized realm create-auth-realm, 36 appclient — launches the Application Client Container and invokes the client application packaged in the application JAR file, 16 asadmin — utility for performing administrative tasks for the Sun Java System Application Server, 18 asant — launches the Jakarta Ant tool, 20 asmigrate — automates migration of J2EE applications from other J2EE platforms to
- automates migration of J2EE applications from other J2EE platforms to Sun Java System Application Server — asmigrate, 22

Sun Java System Application Server, 22

asupgrade — migrates the configuration of a

previously installed Sun Java System

Application Server, 26

C

- capture-schema stores the database metadata (schema) in a file for use in mapping and execution, 28
- checks to see if the JMS provider is up and running jms-ping, 151
- create-acl adds a new access control list file for the named instance, 29
- create-audit-module adds an audit-module, 32
- create-auth-realm adds the new authorized realm, 36
- create-authdb adds the new authorized database for the named instance, 34 create-connector-resource registers the resource with the specified JNDI name, 41

- create-connector-security-map creates a security map for the named connector connection pool, 43
- create-custom-resource registers the custom resource, 45
- create-domain creates a domain with the given name, 47
- create-file-user creates a new file user, 49 create-http-listener adds a new HTTP listener socket, 51
- create-iiop-listener adds the IIOP listener, 55 create-instance — creates an application server instance with the specified instance name, 57
- create-javamail-resource registers the JavaMail resource, 59
- create-jdbc-connection-pool registers the JDBC connection pool, 61
- create-jdbc-resource registers the JDBC resource, 65
- create-jms-resource registers the JMS resource, 70
- create-jmsdest adds the physical destination, 67
- create-jndi-resource registers the JNDI resource, 72
- create-jvm-options creates the JVM options from the Java configuration or profiler elements, 74
- create-persistence-resource registers the persistence resource, 79
- create-profiler creates the profiler element, 81
- create-resource-adapter-config creates the resource adapter javabean, 83
- create-ssl Creates the SSL element in the HTTP listener, IIOP listener, or IIOP Service, 85
- create-virtual-server adds the named virtual server, 89
- creates a domain with the given name create-domain, 47
- creates a new file user create-file-user, 49 creates a security map for the named connector connection pool create-connector-securitymap, 43

- creates an application server instance with the specified instance name create-instance, 57
- creates the JVM options from the Java configuration or profiler elements create-jvm-options, 74
- creates the profiler element create-profiler, 81
- creates the resource adapter javabean create-resource-adapter-config, 83
- Creates the SSL element in the HTTP listener, IIOP listener, or IIOP Service create-ssl, 85

D

- delete-acl removes the access control list file for the named instance, 91
- delete-auth-realm removes the named authorized realm, 95
- delete-authdb removes the authorized database for the named instance, 94
- delete-connector-connection-pool removes the specified connection pool, 96
- delete-connector-resource removes the named connector resource, 98
- delete-custom-resource removes the custom resource, 101
- delete-domain deletes the given domain, 102
- delete-file-user removes the named file user, 103
- delete-http-listener removes the HTTP listener, 105
- delete-http-qos removes the quality of service parameter for the named instance, 106
- delete-iiop-listener removes the IIOP listener, 107
- delete-instance deletes the instance that is not running., 108
- delete-javamail-resource removes the JavaMail resource, 110
- delete-jdbc-connection-pool removes the JDBC connection pool, 111
- delete-jdbc-resource removes the JDBC resource, 113

- delete-jms-resource removes the JMS resource, 116
- delete-jmsdest destroys the physical destination, 114
- delete-jndi-resource removes the JNDI resource, 117
- delete-jvm-options deletes the JVM options from the Java configuration or profiler elements, 118
- delete-lifecycle-module removes the lifecycle module, 120
- delete-mime removes the MIME type for the named instance, 121
- delete-persistence-resource removes the persistence resource, 122
- delete-profiler deletes the profiler element, 123
- delete-resource-adapter-config deletes the resource adapter javabean, 124
- delete-virtual-server deletes the virtual server with the named virtual server ID, 128
- delete-admin-object removes the administered object with the specified JNDI name, 92
- deletes the given domain delete-domain, 102
- deletes the instance that is not running. delete-instance, 108
- deletes the JVM options from the Java configuration or profiler elements delete-jvm-options, 118
- deletes the profiler element delete-profiler, 123
- deletes the resource adapter javabean delete-resource-adapter-config, 124
- deletes the virtual server with the named virtual server ID delete-virtual-server, 128
- deploy deploys the specified component, 129
- deploydir deploys an exploded format of application archive, 133
- deploys an exploded format of application archive deploydir, 133
- deploys the specified component deploy, 129
- destroys the physical destination delete-imsdest, 114

disable — disables the component, 137 display-license — displays the license information, 138 displays a list of all the commands available in the Command-line interface — help, 145 displays the version information — version, 224	gets the MIME types for the named instance — list-mimes, 184 gets the stubs of the client — get-client-stubs, 144 gets the values of the monitorable or configurable attributes — get, 142 gets the virtual servers — list-virtual-servers, 192
E	
enable — enables the component, 139 export — marks a variable name for automatic export to the environment of subsequent commands in multimode, 140	H help — displays a list of all the commands available in the Command-line interface, 145
	1
G	install-license — installs the license file, 150
get — gets the values of the monitorable or configurable attributes, 142	
get-client-stubs — gets the stubs of the	
client, 144	J
gets all the administered objects —	jms-ping — checks to see if the JMS provider is
list-admin-objects, 158	up and running, 151
gets all the connection pools —	jspc — precompiles JSP source files into
list-connector-connection-pools, 163	servlets, 152
gets all the Javamail resources — list-javamail-resources, 175	
gets all the JDBC connection pools —	
list-jdbc-connection-pools, 176	L
gets all the JDBC resources —	launches the Application Client Container and
list-jdbc-resources, 177	invokes the client application packaged in
gets all the JMS resources —	the application JAR file. — appclient, 16
list-jms-resources, 180	launches the Jakarta Ant tool — asant, 20
gets all the JNDI resources —	list — lists the configurable elements, 154
list-jndi-resources, 182	list-acls — gets the access control lists for the
gets all the physical destinations —	named instance, 157
list-jmsdest, 178	list-auth-realms — lists the authorized
gets all the persistence resources —	realms, 161
list-persistence-resources, 185 gets the access control lists for the named	list-components — Lists deployed components, 162
instance — list-acls, 157	list-connector-connection-pools — gets all the
gets the HTTP listeners — list-http-	connection pools, 163
listeners, 172	list-connector-security-maps — lists the security
gets the IIOP listeners — list-iiop-listeners, 173	maps for the connector connection pool, 165
gets the lifecycle modules —	list-domains — lists the domains in the given
list-lifecycle-modules, 183	domains directory, 168

list-file-groups — lists the file groups, 169
list-file-users — lists the file users, 171
list-http-listeners — gets the HTTP
listeners, 172
list-iiop-listeners — gets the IIOP listeners, 173
list-javamail-resources — gets all the Javamail
resources, 175
list-jdbc-connection-pools — gets all the JDBC
connection pools, 176
list-jdbc-resources — gets all the JDBC
resources, 177
list-jms-resources — gets all the JMS
resources, 180
list-jmsdest — gets all the physical
destinations, 178
list-jndi-resources — gets all the JNDI
resources, 182
list-lifecycle-modules — gets the lifecycle
modules, 183
list-mimes — gets the MIME types for the
named instance, 184
list-persistence-resources — gets all the
persistence resources, 185
list-resource-adapter-configs — lists the names
of the configured resource adapters, 188
list-sub-components — Lists EJBs or Servlets in
a deployed module or in a module of a
deployed application, 190
list-virtual-servers — gets the virtual
servers, 192
list-admin-objects — gets all the administered
objects, 158
lists the authorized realms —
list-auth-realms, 161
lists the configurable elements — list, 154
lists the file users — list-file-users, 171
lists the domains in the given domains directory
— list-domains, 168
lists the file groups — list-file-groups, 169
lists the names of the configured resource
adapters — list-resource-adapter-
configs, 188
maps, 165
configs, 188 lists the security maps for the connector connection pool — list-connector-security-

M

marks a variable name for automatic export to the environment of subsequent commands in multimode — export, 140 migrates the configuration of a previously installed Sun Java System Application Server — asupgrade, 26

P

package-appclient — packs the application client container libraries and jar files, 194 packs the application client container libraries and jar files — package-appclient, 194 precompiles JSP source files into servlets — jspc, 152

F

reconfig — applies the changes you have made for a server instance, 198 registers the custom resource create-custom-resource, 45 registers the JavaMail resource create-javamail-resource, 59 registers the JDBC connection pool create-jdbc-connection-pool, 61 registers the JDBC resource create-jdbc-resource, 65 registers the JMS resource create-jms-resource, 70 registers the JNDI resource create-indi-resource, 72 registers the resource in the XML file specified add-resources, 14 registers the resource with the specified JNDI name — create-connector-resource, 41 removes one or more variables from the multimode environment — unset, 217 removes the access control list file for the named instance — delete-acl, 91 removes the administered object with the specified JNDI name delete-admin-object, 92 removes the authorized database for the named instance — delete-authdb, 94

removes the custom resource delete-custom-resource, 101 removes the HTTP listener delete-http-listener, 105 removes the IIOP listener delete-iiop-listener, 107 removes the JavaMail resource delete-javamail-resource, 110 removes the JDBC connection pool delete-jdbc-connection-pool, 111 removes the JDBC resource delete-jdbc-resource, 113 removes the JMS resource delete-jms-resource, 116 removes the JNDI resource delete-indi-resource, 117 removes the lifecycle module delete-lifecycle-module, 120 removes the MIME type for the named instance - delete-mime, 121 removes the named authorized realm delete-auth-realm, 95 removes the named connector resource delete-connector-resource, 98 removes the named file user delete-file-user, 103 removes the persistence resource delete-persistence-resource, 122 removes the quality of service parameter for the named instance — delete-http-qos, 106 removes the specified connection pool delete-connector-connection-pool, 96

S

set — sets the values of attributes, 203 show-component-status — displays the status of the deployed component, 204 shutdown — brings down the administration server and associated instances, 205 start-appserv — starts the domains in the specified or default domains directory, 206 start-domain — starts the given domain, 207 starts the given domain — start-domain, 207 stop-domain — stops the given domain, 211 stops the given domain — stop-domain, 211

stores the database metadata (schema) in a file for use in mapping and execution — capture-schema, 28

U

unset — removes one or more variables from the multimode environment, 217 update-connector-security-map — updates the security map for the named connector connection pool, 218 update-file-user — updates a current file user as specified, 220 updates a current file user as specified — update-file-user, 220 updates the security map for the named connector connection pool — update-connector-security-map, 218 utility for performing administrative tasks for the Sun Java System Application Server — asadmin, 18

٧

validates the J2EE Deployment Descriptors against application server DTDs — verifier, 222
verifier — validates the J2EE Deployment Descriptors against application server DTDs, 222
verifies the content of the domain.xml — verify-domain-xml, 223
verify-domain-xml — verifies the content of the domain.xml, 223
version — displays the version information, 224

W

wscompile — generates stubs, ties, serializers, and WSDL files used in JAX-RPC clients and services, 226
 wsdeploy — reads a WAR file and the jaxrpc-ri.xml file and generates another WAR file that is ready for deployment, 230