



Service Registry 3 2005Q4 Administration Guide

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

Part No: 819-2684-10
October 2005

Copyright 2005 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, Java, NetBeans, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

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

U.S. Government Rights – Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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 2005 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, Java, NetBeans et Solaris sont des marques de fabrique ou des marques déposées, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A 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.



050926@13215



Contents

Preface	7
1 Configuring and Setting Up Service Registry	15
Configuring Service Registry	15
▼ To Configure Service Registry After a Configure Later Installation Using Custom Properties	17
▼ To Enable Use of the Administration Tool	18
Administering the Application Server Domain for Service Registry	18
▼ To Use the Application Server Admin Console	19
▼ To Stop and Restart the Application Server Domain for the Registry	20
▼ To Add Root Certificates to the Trusted Certificates in the Registry Domain	21
Configuring Service Registry for a Non-Default Application Server Installation	22
▼ To Edit the <code>install.properties</code> File	22
Configuring Service Registry for a Non-Default Service Registry Installation	22
▼ To Edit the <code>install.properties</code> File	22
Allowing Access to External Web Sites	23
▼ To Specify a Web Proxy	23
Creating an Administrator	24
▼ To Create an Administrator	24
Configuring the Web Console	25
Adding Predefined Queries	25
▼ To Add a Predefined Query	25
Changing the Default Query	27
▼ To Change the Default Query	27
Hiding Classification Schemes	28

▼ To Hide Classification Schemes	28
Configuring the Search Results Display	29
▼ To Configure the Number of Rows in the Search Results Display	29
▼ To Configure the Columns in the Search Results Area	29
Reinstalling Service Registry	30
▼ To Stop and Delete the Application Server Domain for the Registry	31
▼ To Reinstall the Service Registry Database	31
Backing Up and Restoring the Database	32
▼ To Locate the Apache Derby Documentation	32
2 Using the Administration Tool	33
About the Admin Tool	33
Starting the Admin Tool	34
Batch Mode	34
Interactive Mode	34
Admin Tool Command-line Options	35
Admin Tool Features	36
Permissions	37
Displaying Exceptions	37
Identifying Registry Objects	37
The Effect of Locale on Specifying Names	38
Case Sensitivity	38
Using Admin Tool Commands	38
add association	39
add user	40
cd	46
chown	47
cp	48
echo	49
help	50
lcd	51
ls	51
pwd	53
quit	53
rm	54
select	55
set	56
show	57

users 58

Index 59

Preface

The *Service Registry 3 2005Q4 Administration Guide* describes how to configure Service Registry (“the Registry”) after installation and how to use the administration tool provided with the Registry. This book also describes other administrative tasks, such as backing up and restoring the Registry database.

Who Should Use This Book

The *Administration Guide* is intended for those who need to install, uninstall, and administer the Registry, as well as for those who want to create content for the Registry in bulk rather than use the Web Console to do so.

You should be familiar with the basics of a UNIX[®] command shell environment on your operating system (either the Solaris[™] Operating System or Linux).

Before You Read This Book

Before you read this book, you must install the Registry as described in *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*.

Service Registry is available as part of the Java Web Services Developer Pack (<http://java.sun.com/webservices/jwsdp/>) or as a component of Sun Java Enterprise System, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. If you purchased Service Registry as a component of Java Enterprise System, you should be familiar with the system documentation at <http://docs.sun.com/coll/1286.1>.

Some administrative tasks require you to be familiar with the basic concepts of these specifications:

- *ebXML Registry Information Model Version 3.0*
- *ebXML Registry Services and Protocols Version 3.0*

You can find the latest public versions of these specifications by going to the OASIS web site (<http://www.oasis-open.org/>) and following the links to ebXML RIM V3.0 and ebXML RS V3.0.

How This Book Is Organized

The contents of this book are as follows:

[Chapter 1](#) describes how to configure Service Registry after you install it and how to perform other administrative tasks.

[Chapter 2](#) describes the use of the administration tool.

Service Registry Documentation Set

The Service Registry documentation set is available at <http://docs.sun.com/app/docs/coll/1314.1>. To learn about Service Registry, refer to the books listed in the following table.

TABLE P-1 Service Registry Documentation

Document Title	Contents
<i>Service Registry 3 2005Q4 Release Notes</i>	Contains the latest information about Service Registry, including known problems.
<i>Service Registry 3 2005Q4 Administration Guide</i>	Describes how to configure Service Registry after installation and how to use the administration tool provided with the Registry. It also describes how to perform other administrative tasks.
<i>Service Registry 3 2005Q4 User's Guide</i>	Describes how to use the Service Registry Web Console to search Service Registry and to publish data to it.

TABLE P-1 Service Registry Documentation (Continued)

Document Title	Contents
<i>Service Registry 3 2005Q4 Developer's Guide</i>	Describes how to use the Java API for XML Registries (JAXR) to search Service Registry and to publish data to it.

Related Books

When you install Service Registry, it is deployed to the Sun Java System Application Server. For information about administering Application Server, refer to *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide*.

The Java ES documentation set describes deployment planning and system installation. The URL for system documentation is <http://docs.sun.com/coll/1286.1>. For an introduction to Java ES, refer to the books in the order in which they are listed in the following table.

TABLE P-2 Java Enterprise System Documentation

Document Title	Contents
<i>Sun Java Enterprise System 2005Q4 Release Notes</i>	Contains the latest information about Java ES, including known problems. In addition, components have their own release notes.
<i>Sun Java Enterprise System 2005Q4 Documentation Roadmap</i>	Provides descriptions of all documentation related to Java ES, both as a system and for the individual components.
<i>Sun Java Enterprise System 2005Q4 Technical Overview</i>	Introduces the technical and conceptual foundations of Java ES. Describes components, the architecture, processes, and features.
<i>Sun Java Enterprise System 2005Q4 Deployment Planning Guide</i>	Provides an introduction to planning and designing enterprise deployment solutions based on Java ES. Presents basic concepts and principles of deployment planning and design, discusses the solution life cycle, and provides high-level examples and strategies to use when planning solutions based on Java ES.
<i>Sun Java Enterprise System 2005Q4 Installation Planning Guide</i>	Helps you develop the implementation specifications for the hardware, operating system, and network aspects of your Java ES deployment. Describes issues such as component dependencies to address in your installation and configuration plan.

TABLE P-2 Java Enterprise System Documentation (Continued)

Document Title	Contents
<i>Sun Java Enterprise System 2005Q4 Installation Guide for UNIX</i>	Guides you through the process of installing Java ES on the Solaris Operating System or the Linux operating system. Also shows how to configure components after installation, and verify that they function properly.
<i>Sun Java Enterprise System 2005Q4 Installation Reference</i>	Gives additional information about configuration parameters, provides worksheets to use in your configuration planning, and lists reference material such as default directories and port numbers.
<i>Sun Java Enterprise System 2005Q1 Deployment Example Series: Evaluation Scenario</i>	Describes how to install Java ES on one system, establish a set of core, shared, and networked services, and set up user accounts that can access the services that you establish.
<i>Sun Java Enterprise System 2005Q4 Upgrade Guide</i>	Provides instructions for upgrading Java ES on the Solaris Operating System or the Linux operating environment.
<i>Sun Java Enterprise System Glossary</i>	Defines terms that are used in Java ES documentation.

The URL for all documentation about Java ES and its components is <http://docs.sun.com/prod/entsys.05q4>.

Default Paths and File Names

The following table describes the default paths and file names that are used in this book.

TABLE P-3 Default Paths and File Names

Placeholder	Description	Default Value
<i>ServiceRegistry-base</i>	Represents the base installation directory for Service Registry.	Solaris systems: /opt/SUNWsoar Linux systems: /opt/sun/SUNWsoar
<i>RegistryDomain-base</i>	Represents the directory where the Application Server domain for Service Registry is located and where the Service Registry database is located.	Solaris systems: /var/opt/SUNWsoar Linux systems: /var/opt/sun/SUNWsoar

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P-4 Typographic Conventions

Typeface	Meaning	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>
AaBbCc123	What you type, contrasted with onscreen computer output	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	A placeholder to be replaced with a real name or value	The command to remove a file is <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized (note that some emphasized items appear bold online)	Read Chapter 6 in the <i>User's Guide</i> . <i>A cache</i> is a copy that is stored locally. Do <i>not</i> save the file.

Shell Prompts in Command Examples

The following table shows default system prompts and superuser prompts.

TABLE P-5 Shell Prompts

Shell	Prompt
C shell on UNIX and Linux systems	<code>machine_name%</code>
C shell superuser on UNIX and Linux systems	<code>machine_name#</code>
Bourne shell and Korn shell on UNIX and Linux systems	<code>\$</code>
Bourne shell and Korn shell superuser on UNIX and Linux systems	<code>#</code>
Microsoft Windows command line	<code>C:\</code>

Symbol Conventions

The following table explains symbols that might be used in this book.

TABLE P-6 Symbol Conventions

Symbol	Description	Example	Meaning
[]	Contains optional arguments and command options.	ls [-l]	The -l option is not required.
{ }	Contains a set of choices for a required command option.	-d {y n}	The -d option requires that you use either the y argument or the n argument.
\${ }	Indicates a variable reference.	\${com.sun.javaRoot}	References the value of the com.sun.javaRoot variable.
-	Joins simultaneous multiple keystrokes.	Control-A	Press the Control key while you press the A key.
+	Joins consecutive multiple keystrokes.	Ctrl+A+N	Press the Control key, release it, and then press the subsequent keys.
→	Indicates menu item selection in a graphical user interface.	File → New → Templates	From the File menu, choose New. From the New submenu, choose Templates.

Accessing Sun Resources Online

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

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

- Downloads of Sun products
- Services and solutions
- Support (including patches and updates)
- Training
- Research
- Communities (for example, Sun Developer Network)

Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to <http://docs.sun.com> and click Send Comments. In the online form, provide the full document title and part number. The part number is a 7-digit or 9-digit number that can be found on the book's title page or in the document's URL. For example, the part number of this book is 819-2684.

Configuring and Setting Up Service Registry

This chapter describes how to configure Service Registry after you install it and how to perform other administrative tasks.

This chapter contains the following sections:

- “Configuring Service Registry” on page 15
- “Allowing Access to External Web Sites” on page 23
- “Creating an Administrator” on page 24
- “Configuring the Web Console” on page 25
- “Reinstalling Service Registry” on page 30
- “Backing Up and Restoring the Database” on page 32

Configuring Service Registry

The *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX* describes how to perform post-install configuration of Service Registry using default property settings for the Registry. To use custom property settings, edit the file *ServiceRegistry-base/install/install.properties* before you perform the configuration.

The *ServiceRegistry-base* location is `/opt/SUNWsoar` in the Solaris operating environment and `/opt/sun/SUNWsoar` on Linux systems.

Note – Before you configure Service Registry, you must first install and configure Sun Java System Application Server (“Application Server”). The configuration process for Service Registry installs the Registry into an Application Server domain.

It is recommended that you install Application Server in its default location. If you installed Application Server in a non-default location, follow the instructions in [“Configuring Service Registry for a Non-Default Application Server Installation”](#) on page 22 before you configure Service Registry.

The `install.properties` file contains a set of modifiable property settings. The properties that are listed in [Table 1-1](#) are used by the installation process. Each property name has the prefix `registry.install.` (terminating in a period). Most of these properties set non-default ports for the Application Server domain created for the Registry.

TABLE 1-1 Service Registry Installation Properties

Property Name	Description	Default Property Value
<code>DomainName</code>	Application Server domain name	<code>registry</code>
<code>ServerInstancePort</code>	Application Server HTTP port.	6060
<code>ServerInstanceSecurePort</code>	Application Server HTTPS port.	6443
<code>ServerJMSPort</code>	Application Server Message Queue port	6484
<code>ServerIIOPPort</code>	Application Server IIOP port	6485
<code>ServerIIOPSecurePort</code>	Application Server IIOP secure port	6486
<code>ServerIIOPMutualAuthPort</code>	Application Server IIOP mutual authentication port	6487
<code>AdministrationJMXPort</code>	Application Server JMX port	6488
<code>AdministrationPort</code>	Application Server Administrative Server port	6489
<code>AdministratorUserID</code>	User name used to access Application Server Administrative Server	<code>admin</code>

TABLE 1-1 Service Registry Installation Properties (Continued)

Property Name	Description	Default Property Value
AdministratorPassword	Password used to access Application Server Administrative Server	12345678
ApplicationServerKeystorePassword	Password used to access Application Server keystore	12345678
RegistryServerKeystorePassword	Password used to access Service Registry keystore	12345678

▼ To Configure Service Registry After a Configure Later Installation Using Custom Properties

Before You Begin

To configure the Registry, you must be logged in as root or become superuser.

Steps

1. Change to the *ServiceRegistry-base/install* directory.

2. Edit the modifiable properties in the file *install.properties*.

For security reasons, it is recommended that you not edit this file to change the password values. Instead, specify these values on the command line.

3. After editing the file, run the following command (all on one line):

On Solaris: `/usr/sfw/bin/ant -f build-install.xml install`

On Linux: `/opt/sun/bin/ant --noconfig -f build-install.xml install`

The ant command requires the JAVA_HOME environment variable to be set. Ordinarily, you set this variable to the following value:

```
/usr/jdk/entsys-j2se
```

To specify changed passwords on the command line, specify the following options to the command (all on one line):

```
/usr/sfw/bin/ant -f build-install.xml
-Dregistry.install.RegistryServerKeystorePassword=password1
-Dregistry.install.AdministratorPassword=password2
-Dregistry.install.ApplicationServerKeystorePassword=password3 install
```

The Registry configuration process creates an Application Server domain at *RegistryDomain-base/domains/\${registry.install.DomainName}*. The default domain name is *registry*. The configuration process then starts the domain, deploys the Registry, and leaves the domain running.

The Registry configuration process installs the Registry database and server keystore in the directory *RegistryDomain-base/3.0*. This directory is not removed when the Registry is uninstalled, so that the database can be preserved for use in a future release. The administrator has control over when and whether to remove this directory.

The *RegistryDomain-base* location is */var/opt/SUNWsoar* in the Solaris operating environment and */var/opt/sun/SUNWsoar* on Linux systems.

4. Review the output of the `ant install` command for any errors.

If there are no errors, you can now begin using the Web Console or the Admin Tool.

▼ To Enable Use of the Administration Tool

To perform Admin Tool tasks that are restricted to users with the role of administrator, you need to work around a bug by adding a JAR file to the Admin Tool manifest classpath.

Steps 1. **Make sure you are still in the *ServiceRegistry-base/install* directory.**

2. **Copy the file `soapprocessor.jar` from the deployed Registry to the Registry `lib` directory. Execute the following command (all on one line):**

```
cp
RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/lib/soapprocessor.jar
../lib
```

3. **Run the `admin.jar.manifest.fix` installation target as follows:**

On Solaris: `/usr/sfw/bin/ant -f build-install.xml
admin.jar.manifest.fix`

On Linux: `/opt/sun/bin/ant --noconfig -f build-install.xml
admin.jar.manifest.fix`

Administering the Application Server Domain for Service Registry

The configuration process for Service Registry by default creates an Application Server domain named `registry`, to which the Service Registry web application is deployed. This domain is in the *RegistryDomain-base/domains/registry* directory.

This location is different from the default location for Application Server domains, */var/opt/SUNWappserver/domains* (Solaris) or */var/opt/sun/appserver/domains* (Linux).

To administer the registry domain, you can use the Application Server Administration Console (“Admin Console”). You can use the Admin Console to start and stop the domain, view the server log, and perform other administrative tasks. See “To Use the Application Server Admin Console” on page 19 for details.

You can also examine the server log directly. The log is in the file `RegistryDomain-base/domains/registry/logs/server.log`.

You can also use the `asadmin` command to administer the registry domain. Because the domain is not in the default location, you must specify the `--domaindir` option when you use `asadmin` commands that provide that option.

The password file for the registry domain is `ServiceRegistry-base/pw.txt`. Specify this password file as the argument to the `--passwordfile` option of `asadmin` commands when you administer the registry domain.

The registry domain uses a set of non-default ports so as not to cause conflicts with the default Application Server domain, `domain1`. Table 1–2 lists and describes these ports. For more information, see “Ports in the Application Server” in *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide*.

TABLE 1–2 Service Registry Domain Default Ports

Port Value	Description
6060	HTTP port
6443	HTTPS over SSL
6484	Message Queue port
6485	IIOp port
6486	IIOp SSL port
6487	IIOp Mutual Authentication port
6488	JMX port
6489	Application Server domain administration port

▼ To Use the Application Server Admin Console

- Steps**
1. In a web browser, go to the URL `https://hostname:6489/`.
hostname is the system on which Application Server and Service Registry are running.
 2. Accept the certificate that is offered.
A login page appears
 3. On the login page, type `admin` in the User Name field.

4. Type the Application Server administrator password in the Password field. Use the value that you specified for the `AdministratorPassword` property when you configured the Registry. The default is `12345678`.
5. Click Log In.

See Also For details on using the Admin Console, refer to the online help for the Admin Console or to the *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide*.

To change the logging level for Service Registry, follow the instructions in “To configure log levels” in *Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide*. The property to specify in the Additional Properties area is `org.apache.commons.logging.simplelog.log.org.freebxml.omar`.

▼ To Stop and Restart the Application Server Domain for the Registry

The configuration process for the Registry starts the Application Server domain in which the registry is deployed. After you perform certain administrative tasks, you need to stop and restart the domain. Examples of such tasks are “[Allowing Access to External Web Sites](#)” on page 23 and “[Creating an Administrator](#)” on page 24.

The Admin Console informs you if you need to restart the domain. You can use the Admin Console to perform this task. If you are using the `asadmin` command, you can use Ant tasks to stop and start the domain.

Steps 1. Change to the Service Registry install directory.

```
cd ServiceRegistry-base/install
```

2. Run the following command (all on one line):

```
Solaris: /usr/sfw/bin/ant -f build-install.xml  
appserver.domain.bounce
```

```
Linux: /opt/sun/bin/ant --noconfig -f build-install.xml  
appserver.domain.bounce
```

This target stops the domain and then restarts it.

The `build-install.xml` file also contains separate Ant targets for stopping and starting the Registry domain. To stop the domain, use the Ant target `appserver.domain.stop`. To start the domain, use the Ant target `appserver.domain.start`.

▼ To Add Root Certificates to the Trusted Certificates in the Registry Domain

This task extends the list of trusted certificates in the Application Server registry domain.

Perform this task only if you use a third-party certificate and the root Certificate Authority (CA) certificate for the third party is not already in the Application Server truststore. Do not perform this task if you use only registry-issued certificates.

- Steps**
1. **Download any root certificates that you want to support. Sites that provide root certificates include the following:**
 - <http://www.entrust.net/developer/>
 - http://www.geotrust.com/resources/root_certificates/
 - <http://www.thawte.com/roots/>
 - <http://www.verisign.com/support/roots.html>
 2. **If necessary, use the `unzip` command to extract `.cer` files from the downloaded archive.**

Note – Some files have the suffix `.der`.

3. **Copy the `.cer` files to the directory `ServiceRegistry-base/install/cacerts`.**
4. **Change to the directory `ServiceRegistry-base/install`.**
5. **Run the following command (all on one line):**

```
Solaris: /usr/sfw/bin/ant -f build-install.xml install.cacerts
```

```
Linux: /opt/sun/bin/ant --noconfig -f build-install.xml  
install.cacerts
```

This command installs any certificates found in the directory `ServiceRegistry-base/install/cacerts` into the Application Server domain truststore.

You can use the `list.cacerts` target to make sure that the certificates have been installed correctly.

6. **Follow the instructions in “To Stop and Restart the Application Server Domain for the Registry” on page 20.**

Configuring Service Registry for a Non-Default Application Server Installation

The default location for installing Application Server is `/opt/SUNWappserver/appserver` in the Solaris operating environment and `/opt/sun/appserver` on Linux systems. If you installed Application Server in a different location, you must edit the file `install.properties` before you configure Service Registry.

▼ To Edit the `install.properties` File

- Steps**
1. In the `ServiceRegistry-base/install` directory, open the file `install.properties` in a text editor.
 2. Find the commented-out definition of the property `appserver.root.dir`.
 3. Remove the comment character (`#`) and replace the property definition with the actual location of Application Server.
 4. Save and close the `install.properties` file.

Next Steps Continue with the instructions in [“Configuring Service Registry”](#) on page 15.

Configuring Service Registry for a Non-Default Service Registry Installation

The default location for installing Service Registry is `/opt/SUNWsoar` in the Solaris operating environment and `/opt/sun/SUNWsoar` on Linux systems. If you installed Service Registry in a different location, you must edit the file `install.properties` before you configure Service Registry.

▼ To Edit the `install.properties` File

- Steps**
1. In the `ServiceRegistry-base/install` directory, open the file `install.properties` in a text editor.
 2. Find the commented-out definition of the properties `soar.sdk.home` and `soar.server.home`.
 3. For each property, remove the comment character (`#`) and replace the property definition with the actual location of Service Registry.
 4. Save and close the `install.properties` file.

Next Steps Continue with the instructions in [“Configuring Service Registry”](#) on page 15.

Allowing Access to External Web Sites

Any registry object can have an `ExternalLink` object, which specifies a URL associated with that registry object. In order for users to create `ExternalLink` objects, Service Registry must be able to validate the URL, and this task requires access to external web sites. If the Registry is deployed behind a firewall, you need to set a proxy configuration that allows this access.

Proxy configuration requires you to specify a web proxy host and port as Java Virtual Machine (JVM) options of the Application Server instance where Service Registry is deployed.

▼ To Specify a Web Proxy

Steps 1. Log in to the Application Server Admin Console as described in [“To Use the Application Server Admin Console”](#) on page 19.

2. Expand the Configurations node.

3. Expand the server node, `server-config (Admin Config)`.

4. Click JVM Settings.

5. Click the JVM Options tab.

6. Click Add JVM Option.

7. In the text field, type the following (all on one line):

```
-Dhttp.proxyHost=hostname.domainname -Dhttp.proxyPort=8080
```

The port value is usually 8080. If the port is different in your location, specify the correct value.

8. Click Save.

9. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry”](#) on page 20.

Creating an Administrator

The Service Registry administration tool has some tasks that only a user who is registered as an administrator can perform. In addition, an administrator might be called upon to implement life cycle changes (for example, approvals) to objects other users submit.

An administrator can also change the default access control policy (ACP). However, writing an ACP is currently a manual process that requires knowledge of OASIS eXtensible Access Control Markup Language (XACML). For details, refer to Chapter 9, "Access Control Information Model," of ebXML RIM 3.0, especially the examples in Sections 9.7.6 through 9.7.8. See "Before You Read This Book" on page 7 for information on how to find the ebXML RIM 3.0 specification.

▼ To Create an Administrator

To register yourself as an administrator, follow these steps.

- Steps**
- 1. Perform user registration as described in "Creating a User Account" in *Service Registry 3 2005Q4 User's Guide***
Remember the path name of the certificate you downloaded. The default name of the certificate file is `generated-key.p12`.
 - 2. Obtain the unique identifier of your `User` object as follows:**
 - a. Use the Web Console to perform a Basic Query, with the Object Type set to `User`.**
 - b. Click the Details link to view the `User` object the Registry created for you.**
 - c. Make a note of the Unique Identifier field value.**
 - 3. Copy the certificate to the following location in your home directory, creating directories as needed:**
`$HOME/soar/3.0/jaxr-ebxml/security`
 - 4. Change to the directory**
`RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes.`
 - 5. Open the file `omar.properties` in a text editor.**
 - 6. Find the definition of the property**
`omar.security.authorization.registryAdministrators.`

7. **Edit the property definition by adding a vertical bar (|), followed by the logical identifier string that you made a note of in Step 2.**

The property definition must all be on one line and must not contain spaces. After you finish, it will look something like this (all on one line):

```
omar.security.authorization.registryAdministrators=  
urn:freebxml:registry:predefinedusers:registryoperator|  
urn:uuid:77f5c196-79de-4286-8483-8d80def3583b
```

8. **Save and close the `omar.properties` file.**
9. **Follow the instructions in “To Stop and Restart the Application Server Domain for the Registry” on page 20.**

Next Steps To create additional administrators, you do not have to edit the `omar.properties` file. If you are an administrator, you can use either the Admin Tool or the Web Console to add users, and you can use the Web Console to classify the users as administrators.

Configuring the Web Console

As an administrator, you can customize some aspects of the Web Console display by editing configuration files. This section describes the following tasks:

- “Adding Predefined Queries” on page 25
- “Changing the Default Query” on page 27
- “Hiding Classification Schemes” on page 28
- “Configuring the Search Results Display” on page 29

For information about using the Web Console, see the *Service Registry 3 2005Q4 User’s Guide*.

Adding Predefined Queries

Service Registry includes several predefined queries, which appear in the Web Console Search form in the Select Predefined Query drop-down list. As an administrator, you can add new queries to the drop-down list that are specific to your installation of the Registry.

▼ To Add a Predefined Query

- Steps** 1. **Use the Web Console to publish an `AdhocQuery` object to the Registry.**

The name and description you specify for the query will appear in the drop-down list of predefined queries. In the SQL statement for the query, specify placeholders

for user-supplied data by enclosing them in pairs of single quotes, as follows:

```
select * from registryobject where id = '$lid'
```

2. Make a note of the unique identifier of the `AdhocQuery` object and of any placeholders in the SQL statement.
3. Change to the directory `RegistryDomain-base/3.0/jaxr-ebxml`.
4. Open the file `registry-browser-config.xml` in a text editor.
5. Add an entry to the `registry-browser-config.xml` file, using the following format. Specify a `Parameter` element for each placeholder in the SQL statement.

```
<Query>
  <AdhocQueryRef id="unique_identifier" />
  <Parameter parameterName="$placeholder_name" datatype="string">
    <rim:Name>
      <rim:LocalizedString xml:lang="en" charset="UTF-8"
        value="parameter_name_in_en_locale" />
      <rim:LocalizedString xml:lang="fr" charset="UTF-8"
        value="parameter_name_in_fr_locale" />
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString xml:lang="en" charset="UTF-8"
        value="parameter_description_in_en_locale" />
      <rim:LocalizedString xml:lang="fr" charset="UTF-8"
        value="parameter_description_in_fr_locale" />
    </rim:Description>
  </Parameter>
  ...
</Query>
```

The `unique_identifier` is the unique identifier of the `AdhocQuery` object.

The `parameterName` attribute value for each parameter must come from a placeholder in the SQL statement for the query.

The `datatype` attribute can have any of the following values:

- `string`: The parameter appears as a text field in the Search form.
- `taxonomyElement`: The parameter appears as a drop-down list in the Search form. If you specify a `taxonomyElement` data type, the Name and Description elements must be followed by a `SlotList` element that looks like this:

```
<rim:SlotList>
  <rim:Slot name="domain">
    <rim:ValueList>
      <rim:Value>
        classification_scheme_or_concept_id
      </rim:Value>
    </rim:ValueList>
  </rim:Slot>
</rim:SlotList>
```

The *classification_scheme_or_concept_id* is the unique identifier of the classification scheme or concept whose concepts (or subconcepts) must appear in the drop-down list. You must publish the classification scheme if it does not already exist in the registry.

The slot name must be "domain".

- `boolean`: The parameter appears as a checkbox in the Search form.

If the datatype is `string` or `boolean`, you can also add a `defaultValue` attribute to the `Parameter` element to specify a default value to appear in the Search form.

Specify localized string values for each parameter name and description for any locales you support. The *parameter_name* in the current locale appears as the label of the parameter in the Search form.

Use the existing entries in the `registry-browser-config.xml` file as a reference.

6. Save and close the `registry-browser-config.xml` file.
7. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry” on page 20](#).

Changing the Default Query

The query that appears as the default in the Select Predefined Query drop-down list is Basic Query, which allows users to search for registry objects by name, description, and classification.

As an administrator, you can change this default to a query that is appropriate to your installation. For example, you might want the default query to be a new predefined query that you added to the Registry, as described in [“Adding Predefined Queries” on page 25](#). To make this change, edit a property in a configuration file.

▼ To Change the Default Query

- Steps**
1. Change to the directory
`RegistryDomain-base/domains/registry/applications/j2ee-modules/
soar/WEB-INF/classes`.
 2. Open the file `jaxr-ebxml.properties` in a text editor.
 3. Find the definition of the property `jaxr-ebxml.thin.defaultQueryPanel`. By default, this property is commented out:

```
#jaxr-ebxml.thin.defaultQueryPanel=
```

4. Remove the comment character (#).
5. Set the value of the property by specifying the logical identifier of the query that will be the default, as in the following example:


```
jaxr-ebxml.thin.defaultQueryPanel=urn:oasis:names:tc:ebxml-regrep:query:MyQuery
```
6. Save and close the `jaxr-ebxml.properties` file.
7. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry”](#) on page 20.

Hiding Classification Schemes

A tree structure of classification schemes appears in the following areas of the Web Console:

- The Search form area, when Basic Query is selected
- The Explore menu area
- The ClassificationScheme/Concept Selector window that appears when you need to choose a concept for some kinds of registry objects

As an administrator, you can hide classification schemes from view if you do not want the classification schemes to be available to users of Service Registry. To hide classification schemes, define a property in a configuration file.

▼ To Hide Classification Schemes

- Steps**
1. Change to the directory
`RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes`.
 2. Open the file `jaxr-ebxml.properties` in a text editor.
 3. Set the property
`jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList` by using the following syntax. All of the property definition must be on one line and must not contain spaces.

```
jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList=
class_scheme_id1|class_scheme_id2|...
```

Specify the logical identifier of each classification scheme that is to be hidden. If you specify more than one identifier, separate the identifiers with a vertical bar (|), as in the following example:

```
jaxr-ebxml.registryBrowser.ConceptsTreeModel.hiddenSchemesList=
urn:oasis:names:tc:ebxml-regrep:classificationScheme:StatusType|
urn:oasis:names:tc:ebxml-regrep:profile:ws:classificationScheme:BindingType
```

4. Save and close the `jaxr-ebxml.properties` file.
5. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry” on page 20.](#)

Configuring the Search Results Display

By default, the Web Console displays 10 search results at a time for each query. If the search returns more than 10 results, users can display additional pages of results. As an administrator, you can modify the number of search results that appears on each page.

By default, the Web Console displays certain columns in the search results area. For each object, it displays the object type, name, description, version, and version comment. For some object types, a non-default display is configured. For example, for a `ServiceBinding` object, the display includes the endpoint instead of the version information. As an administrator, you can add configuration information to display non-default data for object types of your choice.

To perform each of these tasks, you edit a configuration file.

▼ To Configure the Number of Rows in the Search Results Display

- Steps**
1. Change to the directory `RegistryDomain-base/domains/registry/applications/j2ee-modules/soar/WEB-INF/classes`.
 2. Open the file `jaxr-ebxml.properties` in a text editor.
 3. Find the definition of the property `omar.client.thinbrowser.numSearchResults`:

```
omar.client.thinbrowser.numSearchResults=10
```
 4. Change the value 10 to the value you prefer.
 5. Save and close the `jaxr-ebxml.properties` file.
 6. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry” on page 20.](#)

▼ To Configure the Columns in the Search Results Area

- Steps**
1. Change to the directory `RegistryDomain-base/3.0/jaxr-ebxml`.
 2. Open the file `registry-browser-config.xml` in a text editor.

3. Add an entry to the registry-browser-config.xml file, using the following format.

This example configures a non-default display for Service objects.

```
<ObjectTypeConfig
  className="org.freebxml.omar.client.xml.registry.infomodel.ServiceImpl"
  id="urn:oasis:names:tc:ebxml-regrep:ObjectType:RegistryObject:Service">
  <SearchResultsConfig>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Object Type" columnWidth="25" editable="false"
      method="getObjectType"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Name" columnWidth="25" editable="true" method="getName"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Description" columnWidth="30" editable="true"
      method="getDescription"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Status" columnWidth="15" method="getStatusAsString"/>
    <SearchResultsColumn columnClass="java.lang.Object"
      columnHeader="Version" columnWidth="5" method="getVersionName"/>
  </SearchResultsConfig>
</ObjectTypeConfig>
```

The registry-browser-config.xml provides syntax for the ObjectTypeConfig element. Use the elements that are already in the file as examples. These elements configure the default display for registry objects as well as non-default displays for ExternalLink, ExtrinsicObject, and ServiceBinding objects.

For the most part, you can deduce the method names from the class attributes in the *ebXML Registry Information Model Version 3.0* specification (see [“Before You Read This Book” on page 7](#) for details). The getStatusAsString method can be found in the RegistryObjectImpl implementation class. (This release of Service Registry does not include API documentation, however.)

4. Save and close the registry-browser-config.xml file.
5. Follow the instructions in [“To Stop and Restart the Application Server Domain for the Registry” on page 20](#).
6. To verify the reconfiguration, use the Search or Explore menu of the Web Console to display the objects whose columns you changed.

Reinstalling Service Registry

If you need to uninstall and reinstall Service Registry, perform the following tasks before you reinstall:

- If the Registry database contains data that you want to preserve, back up the database as described in [“Backing Up and Restoring the Database” on page 32](#).
- Stop the Application Server domain for the Registry, then delete the domain. If you do not delete the domain, post-install configuration of the reinstalled Registry will fail.

If you need to reinstall the Service Registry database (for example, if the database becomes corrupted), follow the instructions in [“To Reinstall the Service Registry Database” on page 31](#). You do not need to uninstall the database before you reinstall it.

▼ To Stop and Delete the Application Server Domain for the Registry

Steps 1. Change to the *ServiceRegistry-base/install* directory.

2. Run the following command (all on one line):

```
Solaris: /usr/sfw/bin/ant -f build-install.xml
appserver.domain.stop
```

```
Linux: /opt/sun/bin/ant --noconfig -f build-install.xml
appserver.domain.stop
```

3. Run the following command (all on one line):

```
Solaris: /usr/sfw/bin/ant -f build-install.xml
appserver.domain.delete
```

```
Linux: /opt/sun/bin/ant --noconfig -f build-install.xml
appserver.domain.delete
```

▼ To Reinstall the Service Registry Database

Steps 1. Change to the *ServiceRegistry-base/install* directory.

2. Run the following command (all on one line):

```
Solaris: /usr/sfw/bin/ant -f build-install.xml install.db
```

```
Linux: /opt/sun/bin/ant --noconfig -f build-install.xml
install.db
```

Backing Up and Restoring the Database

The Registry uses the Apache Derby database. By default, the database is located in the directory *RegistryDomain-base/3.0/data/registry/soar/*.

To learn how to back up and restore the database, consult the Apache Derby documentation.

▼ To Locate the Apache Derby Documentation

- Steps**
1. In a web browser, go to the Apache Derby web site (<http://db.apache.org/derby/>).
 2. Click the Manuals tab.
 3. Click 10.0 Manuals.
 4. Locate the Server & Admin Guide.
 5. Locate the sections on backing up and restoring databases.

Using the Administration Tool

This chapter describes how to use the Administration Tool (“the Admin Tool”) for the Service Registry.

This chapter contains the following sections:

- “About the Admin Tool” on page 33
- “Starting the Admin Tool” on page 34
- “Admin Tool Features” on page 36
- “Using Admin Tool Commands” on page 38

About the Admin Tool

The Service Registry Administration Tool provides a simple command-line interface for common administration tasks, such as adding associations to the Registry and removing objects from the Registry.

The tool can operate in either of two modes:

- In batch mode, you specify one or more commands on the tool’s command line.
- In interactive mode, you enter commands in the tool’s interactive shell.

Several commands, such as `ls` and `rm`, mimic both the name and the behavior of well-known UNIX[®] commands that operate on files and folders. Other commands have no corresponding UNIX equivalent.

Starting the Admin Tool

To start the Admin Tool, you execute the `admin-tool.jar` file:

```
java -jar ServiceRegistry-base/lib/admin-tool.jar [options]...
```

The `ServiceRegistry-base` location is `/opt/SUNWsoar` in the Solaris operating environment and `/opt/sun/SUNWsoar` on Linux systems.

Ignore the warnings that appear when you start the tool; they are not significant.

To exit the Admin Tool, use the `quit` command.

To perform commands restricted to administrators, you need to specify the type, location, and password of your certificate file (that is, your keystore) on the command line as follows (all on one line):

```
java -Djaxr-ebxml.security.storetype=PKCS12 \  
-Djaxr-ebxml.security.keystore=security/filename.p12 \  
-Djaxr-ebxml.security.storepass=password -jar admin-tool.jar
```

Here, `filename` is the root name of your certificate file (by default, `generated-key`), and `password` is the password you specified to protect this file when you created your certificate. The location `security/filename.p12` is relative to the directory `$HOME/soar/3.0/jaxr-ebxml`.

To save typing, create a script to execute this command.

Batch Mode

To run the Admin Tool in batch mode, specify the `-command` option on the command line when you start the Admin Tool.

For example, the following command executes the `ls` command:

```
java -jar ServiceRegistry-base/lib/admin-tool.jar -command "ls *.html"
```

The Admin Tool echoes your commands and the tool's responses to the screen and then exits after your commands have been executed.

Make sure that you properly escape any characters that are significant to your shell.

Interactive Mode

To run the Admin Tool in interactive mode, start the Admin Tool shell by specifying any options other than `-command` (or no options) on the command line:

```
java -jar ServiceRegistry-base/lib/admin-tool.jar
```

The Admin Tool displays the following prompt and waits for your input:

```
admin>
```

Admin Tool Command-line Options

The Admin Tool recognizes the command-line options that are listed in “Synopsis” on page 35 and described in “Options” on page 35.

Synopsis

```
[-alias alias] [-command commands] [-debug] [-help] [-keypass keypass]  
[-localdir localdir] [-locale locale] [-registry url]  
[-root locator [-create]] [-sqlselect SQL_statement] [-verbose]
```

Options

- | | |
|----------|---|
| -alias | The alias to use when accessing the user’s certificate in the keystore. Specify the alias that you used when you registered as a user. |
| -command | <p>The Admin Tool command sequence to run instead of getting commands from the interactive shell. Use a semicolon (;) to separate multiple commands. You do not have to include a quit command in <i>commands</i>. If you need to use a semicolon that is not a command separator, precede the semicolon by a backslash:</p> <pre>\;</pre> <p>The shell in which you run the Admin Tool might require you to escape the backslash with a second backslash:</p> <pre>\\;</pre> <p>If any command contains spaces, enclose the entire command sequence in single or double quotes so that the tool will treat the sequence as one command-line parameter instead of several. If your shell also interprets a semicolon as separating shell commands, you always have to put sequences of multiple Admin Tool commands in quotation marks.</p> |
| -create | If necessary, create the RegistryPackage specified by the -root option as well as any parent RegistryPackage objects as needed. This option is valid only if the user who is running the Admin Tool is authorized to create objects. |
| -debug | Outputs extra information that is useful when debugging. |

-help	Provides a list of these options.
-keypass	The password to use when accessing a user's certificate in the keystore. Specify the password that you used when you registered as a user.
-localdir	The base directory in the local file system for commands that relate to files in the local file system.
-locale	The locale (for example, en or fr) to use for selecting the resource bundle to use for error and status messages. The default is determined by the Java Virtual Machine (JVM).
-registry	The URL of the ebXML registry to which to connect. The default is <code>http://localhost:6060/soar/registry/soap</code> .
-root	The locator (for example, <code>/registry/userData</code>) of the <code>RegistryPackage</code> to use as the base for those commands that treat the repository as a tree of <code>RegistryPackage</code> objects that each contain other <code>RegistryObject</code> and <code>RegistryPackage</code> objects. The default is the <code>RegistryPackage</code> that is defined for all users' data: <code>/registry/userData</code> .
-sqlselect	Execute <i>SQL_statement</i> to select registry objects. The statement should be a complete SQL statement that starts with <code>select</code> . The SQL statement must be enclosed in quotation marks, but it does not have to be terminated by a semicolon. If you specify this option and then use the <code>select</code> command with no argument, the command will execute <i>SQL_statement</i> until you use the <code>select</code> command with an argument other than <i>SQL_statement</i> .
-v -verbose	Specifies the verbose output of status messages.

Note – The output of the `-help` option lists two options that are not supported in this release: `-class` and `-property`.

Admin Tool Features

This section describes the following features of the Admin Tool:

- “Permissions” on page 37
- “Displaying Exceptions” on page 37
- “Identifying Registry Objects” on page 37
- “The Effect of Locale on Specifying Names” on page 38
- “Case Sensitivity” on page 38

Permissions

When you use the Admin Tool, you can perform only those actions that are allowed for the user whose key alias and password you specified when you started the tool. Only a user with the role of administrator can perform certain commands (`chown`, for example). See “Creating an Administrator” on page 24 for details.

Displaying Exceptions

The Admin Tool enables you to avoid viewing long stack traces when a command fails.

When a command fails, the Admin Tool prints the first line of the stack trace and the following message:

```
An error occurred when executing the function. Use the show exception
command to view messages.
```

If you need more information, execute the `show exception` command next to see the full stack trace.

The `show exception` command always displays the stack trace of the immediately preceding command.

Identifying Registry Objects

The primary way to identify registry objects is by name. However, you normally identify `RegistryPackage` objects by the path from the `registry` root to the `RegistryPackage`. For example, `/registry/userData` is the path to the `userData RegistryPackage`.

Some matches for names support wildcards. Use a question mark (?) to match a single character. Use an asterisk (*) to match zero or more characters.

Some commands (for example, `cd` and `chown`) support identifying objects by their Uniform Resource Name (URN), which must include a leading `urn:`. For example, `urn:uuid:2702f889-3ced-4d49-82d1-e4cd846cb9e4` is a valid URN.

The `chown` command also supports the use of `%number` to refer to a `User` listed by a previous `users` command.

For some commands, you can enter names that contain spaces by enclosing the entire name in double quotes or by preceding each space in the name by a backslash.

The `select` command supports the use of SQL wildcards: the percent sign (%) to match multiple characters, and the underscore (_) to match a single character.

The Effect of Locale on Specifying Names

A `RegistryObject` (or a `RegistryPackage`) can have multiple names, each of which is associated with a different locale.

The paths and object names that you specify are evaluated with respect to the current locale only. When you attempt to select by name a registry object that has multiple names, the Registry attempts to match the name that you provide against only one alternative for the registry object's name (the choice whose locale most closely matches the current locale), not against all the multiple names for the registry object.

For example, suppose the current `RegistryPackage` has a member object that has two names, each associated with a different locale: `red` in the `en` (English) locale and `rouge` in the `fr` (French) locale. When the current locale is `en`, the command `ls rouge` does not display that member object, but when the locale is `fr` (or one of its variants), it does.

Case Sensitivity

Command names and literal parameters that are recognized by the Admin Tool are not case sensitive. For example, `ls`, `Ls`, and `LS` are equivalent.

Options to which you provide the value are passed literally to the code that uses the option.

Using Admin Tool Commands

The following sections describe the available commands. For each command, the synopsis and the descriptions of the options and operands observe the following typographical conventions:

- *Italics* indicate an option argument or operand that should be replaced by an actual value when you run the command.
- Curly braces (`{ }`) delimit a choice of options or operands where you must include one of the options or operands. The options or operands are separated by a vertical bar (`|`).
- Square brackets (`[]`) delimit an option or operand, or a choice of options or operands, that may be omitted.
- An ellipsis (`. . .`) after an option or operand indicates that you may repeat the argument or operand.

Anything else is literal text that you must include when running the command.

add association

Adds an Association object to the Registry.

Synopsis

```
add association -type association-type sourceURN targetURN
```

Description

The `add association` command adds an Association object of the specified type to the Registry. You can use any of the following types:

- `AccessControlPolicyFor`
- `AffiliatedWith` (which has the subconcepts `EmployeeOf` and `MemberOf`)
- `Contains`
- `ContentManagementServiceFor`
- `EquivalentTo`
- `Extends`
- `ExternallyLinks`
- `HasFederationMember`
- `HasMember`
- `Implements`
- `InstanceOf`
- `InvocationControlFileFor` (which has the subconcepts `CatalogingControlFileFor` and `ValidationControlFileFor`)
- `OffersService`
- `OwnerOf`
- `RelatedTo`
- `Replaces`
- `ResponsibleFor`
- `SubmitterOf`
- `Supersedes`
- `Uses`

Options

`-type` The type of the Association object.

Operands

`sourceURN` The URN of the source object.

`targetURN` The URN of the target object.

Example

The following command (all on one line) creates a `RelatedTo` relationship between the objects with the two specified URNs.

```
admin> add association -type RelatedTo
urn:uuid:ab80d8f7-3bea-4467-ad26-d04a40045446
urn:uuid:7a54bbca-2131-4a49-8ecc-e7b4ac86c4fd
```

add user

Adds a user to the Registry.

Synopsis

```
add user [-edit] [-load filename] [-firstname string] [-lastname string]
[-middleName string] -alias string -keypass string [-post1.type string]
[-post1.city string] [-post1.country string] [-post1.postalcode string]
[-post1.stateOrProvince string] [-post1.street string]
[-post2.streetNumber string] [-post2.type string] [-post2.city string]
[-post2.country string] [-post2.postalcode string]
[-post2.stateOrProvince string] [-post2.street string]
[-post2.streetNumber string] [-post3.type string] [-post3.city string]
[-post3.country string] [-post3.postalcode string]
[-post3.stateOrProvince string] [-post3.street string]
[-post3.streetNumber string] [-telephone1.type string]
[-telephone1.areaCode string] [-telephone1.countryCode string]
[-telephone1.extension string] [-telephone1.number string]
[-telephone1.URL string] [-telephone2.type string]
[-telephone2.areaCode string] [-telephone2.countryCode string]
[-telephone2.extension string] [-telephone2.number string]
[-telephone2.URL string] [-telephone3.type string]
[-telephone3.areaCode string] [-telephone3.countryCode string]
```



```
[-telephone3.extension string] [-telephone3.number string]  
[-telephone3.URL string] [-email1.type string] [-email1.address string]  
[-email2.type string] [-email12address string] [-email3.type string]  
[-email3.address string]
```

Description

The `add user` command adds a `User` object. A `User` object normally contains at least one `PostalAddress`, `PhoneNumber`, and `EmailAddress` object.

Specify the information about the user either on the command line itself or by using the `-load` option to specify a Java property file with the information. The information options and the `-load` option are evaluated in the order in which they appear on the command line. For example, you can specify some properties on the command line, load others from a property file, and then override information in the property file with later command-line options

You can specify up to three addresses, telephone numbers, and email addresses for a new user. If you need more, you can add them later using the Web Console or JAXR.

When you specify an address, telephone number, or email address, you must provide a value for its type: for example, `-emailType OfficeEmail`.

You can use shorthand options (such as `-fn`) on the command line for some of the common information that is required for every user. However, you must use the longer form when you provide the information in a property file. For example, you can specify the user's first email address on the command line using either `-email1.address`, `-emailAddress`, or `-email`. However, when you specify the first email address in a property file, you must use `email1.address=`. Because there is only one option for the user's second email address, you must use `-email2.address` on the command line and `email2.address=` in a property file.

If you specify the `-edit` option, the Admin Tool launches an editor so that you can edit the new user's information. See the option description for details.

Note – The property files that you load with `-load` or edit with `-edit` use the ISO-8859-1 charset, as do all Java property files. See the documentation for `java.util.Properties.load(InputStream)` for details on how to represent characters not in ISO-8859-1 in property files.

Options

`-edit`

Causes the Admin Tool to launch an editor so that you can edit the new user's information. The tool launches the editor after evaluating the other command-line parameters. Therefore, editing starts with the result of evaluating any information that was specified on the command line or in a property file. The editing program must terminate without error before the command can continue. The Admin Tool launches the editor specified by the `set editor` command (see "[set](#)" on page 56); by default, this is the `vi` editor.

Note – At this release, `-edit` works with `emacsclient` and the NetBeans™ command `bin/runide.sh --open` (but not very well), and has not been shown to work with `vi`.

`-load`

Specifies a Java property file whose contents specify properties for the user. The property names are the same as the long forms of the `add user` command options (for example, `lastName` and `post1.type`).

`-fn | -firstName`

Specifies the first name of a user.

`-ln | -lastName`

Specifies the last name (surname) of a user. The last name, which is required, must be specified either on the command line or in a property file.

`-mn | -middleName`

Specifies the middle name of a user.

`-alias`

The alias to use when accessing the user's certificate in the keystore. This option is required. The alias must be at least three characters long.

`-keypass`

The password to use when accessing a user's certificate in the keystore. This option is required. The password must be at least six characters long.

`-postalType | -post1.type`

The type of the first `PostalAddress`. The type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string (for example, `Office` or `Home`).

`-city | -post1.city`

The city of the first `PostalAddress`.

`-country | -post1.country`

The country of the first `PostalAddress`.

`-postalCode | -postcode | -zip | -post1.postalcode`

The postal code of the first `PostalAddress`.

-stateOrProvince | -state | -province | -post1.stateOrProvince
The state or province of the first PostalAddress.

-street | -post1.street
The street name of the first PostalAddress.

-streetNumber | -number | --post1.streetNumber
The street number of the first PostalAddress.

-post2.type
The type of the second PostalAddress. If a second PostalAddress is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string (for example, Office or Home).

-post2.city
The city of the second PostalAddress.

-post2.country
The country of the second PostalAddress.

-post2.postalcode
The postal code of the second PostalAddress.

-post2.stateOrProvince
The state or province of the second PostalAddress.

-post2.street
The street name of the second PostalAddress.

-post2.streetNumber
The street number of the second PostalAddress.

-post3.type
The type of the third PostalAddress. If a third PostalAddress is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string (for example, Office or Home).

-post3.city
The city of the third PostalAddress.

-post3.country
The country of the third PostalAddress.

-post3.postalcode
The postal code of the third PostalAddress.

-post3.stateOrProvince
The state or province of the third PostalAddress.

-post3.street
The street name of the third PostalAddress.

-post3.streetNumber
The street number of the third PostalAddress.

- phoneType | -telephone1.type
The type of the first TelephoneNumber. The type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: Beeper, FAX, HomePhone, MobilePhone, or OfficePhone.
- areaCode | -telephone1.areaCode
The area code of the first TelephoneNumber.
- countryCode | -telephone1.countryCode
The country code of the first TelephoneNumber.
- extension | -telephone1.extension
The extension of the first TelephoneNumber.
- number | -telephone1.number
The telephone number suffix, not including the country or area code, of the first TelephoneNumber. The number, which is required, must be specified either on the command line or in a property file.
- URL | -telephone1.URL
The URL of the first TelephoneNumber (the URL that can dial this number electronically).
- telephone2.type
The type of the second TelephoneNumber. If a second TelephoneNumber is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: Beeper, FAX, HomePhone, MobilePhone, or OfficePhone.
- telephone2.areaCode
The area code of the second TelephoneNumber.
- telephone2.countryCode
The country code of the second TelephoneNumber.
- telephone2.extension
The extension of the second TelephoneNumber.
- telephone2.number
The telephone number suffix, not including the country or area code, of the second TelephoneNumber. If a second TelephoneNumber is specified, the number, which is required, must be specified either on the command line or in a property file.
- telephone2.URL
The URL of the second TelephoneNumber (the URL that can dial this number electronically).

- telephone3.type
The type of the third TelephoneNumber. If a third TelephoneNumber is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: Beeper, FAX, HomePhone, MobilePhone, or OfficePhone.
- telephone3.areaCode
The area code of the third TelephoneNumber.
- telephone3.countryCode
The country code of the third TelephoneNumber.
- telephone3.extension
The extension of the third TelephoneNumber.
- telephone3.number
The telephone number suffix, not including the country or area code, of the third TelephoneNumber. If a third TelephoneNumber is specified, the number, which is required, must be specified either on the command line or in a property file.
- telephone3.URL
The URL of the third TelephoneNumber (the URL that can dial this number electronically).
- emailType | -email1.type
The type of the first EmailAddress. The type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: HomeEmail or OfficeEmail.
- emailAddress | -email | -email1.address
The first email address. The first email address is required.
- email2.type
The type of the second EmailAddress. If a second EmailAddress is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: HomeEmail or OfficeEmail.
- email2.address
The second email address.
- email3.type
The type of the third EmailAddress. If a third EmailAddress is specified, the type, which is required, must be specified either on the command line or in a property file. The value is an arbitrary string, but you can specify one of the following known types: HomeEmail or OfficeEmail.
- email3.address
The third email address.

Examples

The following command loads the `User` properties from the file `JaneSmith.properties` in the user's home directory.

```
admin> add user -load ~/JaneSmith.properties
```

The following command (all on one line) specifies the minimum properties that are required to create a `User` object.

```
admin> add user -ln Smith -postaltype Office -country US  
-phonetype Office -number 333-3333 -emailtype OfficeEmail  
-emailaddress JaneSmith@JaneSmith.com -alias 123 -keypass 123456
```

cd

Changes the `RegistryPackage` location.

Synopsis

```
cd {locator | URN}
```

Description

The `cd` command changes directory (metaphorically) to the `RegistryPackage` at the specified path or with the specified URN.

The command changes to a specified URN when multiple `RegistryPackage` objects exist with the same path (for the current locale).

Operands

locator The path of names of registry objects from the root of the repository to an object in the repository, with each name preceded by a forward slash (/).

For example, the locator for the `userData` `RegistryPackage` that is a member of the registry `RegistryPackage` (which is not itself a member of any `RegistryPackage`) is `/registry/userData`. The locator for the `folder1` `RegistryPackage` that is a member of the `userData` `RegistryPackage` is `/registry/userData/folder1`.

URN The URN of the `RegistryPackage`, which must be a URN starting with `urn:.`

Examples

The following command changes the directory to the RegistryPackage with the URN `urn:uuid:92d3fd01-a929-4eba-a5b4-a3f036733017`.

```
admin> cd urn:uuid:92d3fd01-a929-4eba-a5b4-a3f036733017
```

The following command changes the directory to the location `/registry/userData/myData`.

```
admin> cd /registry/userData/myData
```

chown

Changes the owner of a RegistryObject.

Synopsis

```
chown {URN | %index}
```

Description

The `chown` command changes the ownership of the objects selected with a preceding `select` command to the user specified either by the URN or by the reference to the user's URN that was listed by a preceding `users` command.

Only a user with the role of administrator can execute this command successfully.

Operands

URN The User object specified by the URN.

%*index* A numerical reference to a URN for a User object listed in a preceding `users` command.

Examples

The following command changes the ownership of the selected objects to the user specified by the URN `urn:uuid:26aa17e6-d669-4775-bfe8-a3a484d3e079`.

```
admin> chown urn:uuid:26aa17e6-d669-4775-bfe8-a3a484d3e079
```

The following command changes the ownership of the selected objects to the user with the number 2 in a preceding `users` command.

```
admin> chown %2
```

cp

Copies files and folders into the Registry.

Synopsis

```
cp [-owner {URN | %index}] [-exclude pattern]... [-include pattern]...  
pattern...
```

Description

The `cp` command copies folders and files into the Registry as `RegistryPackage` and `ExtrinsicObject` objects, respectively.

The local directory on the local file system from which to copy files and folders defaults to the current directory from which you started the Admin Tool. You can use the `-localdir` option to change the local directory when you start the Admin Tool. You can use the `lcd` command to change the local directory after the Admin Tool has started. You can get the absolute path of the current local directory by using the `show localdir` command.

The command is recursive. That is, if you specify a directory, the command copies all the files and folders under the directory.

Options

- owner Sets the owner of the copied registry objects to the user specified by the *URN* or *%index* argument. See the description of the `chown` command for a description of these arguments. You must have the role of administrator to specify an owner other than yourself.
- exclude Copies all files except those whose names contain the specified pattern, where *pattern* is a pattern comprising literal characters and the special characters asterisk (*) (representing zero or more characters) and question mark (?) (representing one and only one character).

You can specify this option more than once.

`-include` Copies all files whose names contain the specified pattern, where *pattern* is a pattern comprising literal characters and the special characters asterisk (*) (representing zero or more characters) and question mark (?) (representing one and only one character).

You can specify this option more than once.

Operands

pattern The files or folders to be copied, specified by a pattern comprising literal characters and the special characters asterisk (*) (representing zero or more characters) and question mark (?) (representing one and only one character). You can specify more than one *pattern*.

Examples

The following command copies the directory `mydir` to the Registry, to be owned by the user with the number 4 in a preceding `users` command.

```
admin> cp -owner %4 mydir
```

The following command copies the directory `mydir` to the Registry, excluding files and directories that end with the string `.z` or `.c`.

```
admin> cp mydir -exclude *.z -exclude *.c
```

echo

Echoes a string.

Synopsis

```
echo string
```

Description

The `echo` command echoes the specified *string* to the output. This command is most useful when you specify it in the `-command` option when you run the Admin Tool in batch mode.

Operand

string A sequence of characters.

Example

The following command prints the date and the result of the `ls` command into a log file.

```
java -jar admin-tool.jar -command "echo "date"; ls" > admin.log
```

help

Displays information about commands.

Synopsis

```
help [command_name]
```

Description

The `help` command displays information about the available commands or a specified command.

For commands with subcommands, such as `add` and `show`, the `help` command displays information about the subcommands.

If you do not specify an argument, the `help` command displays usage information for all commands.

Operand

command_name The name of an Admin Tool command.

Examples

The following command displays usage information for all commands.

```
admin> help
```

The following command displays usage information for the `lcd` command.

```
admin> help lcd
```

The following command displays usage information for the add subcommands.

```
admin> help add
```

lcd

Changes the current directory on the local file system.

Synopsis

```
lcd [path_name]
```

Description

The `lcd` command changes the current local directory on the local file system.

If you do not specify an argument, the `lcd` command changes the current directory to your default home directory.

Operand

path_name A directory name, which can be absolute or relative.

Examples

The following command changes the current local directory to the `/usr/share` directory.

```
admin> lcd /usr/share
```

The following command changes the current local directory to your default home directory on the local file system.

```
admin> lcd
```

ls

Lists the objects in the current RegistryPackage.

Synopsis

`ls [{pattern | URN}...]`

Description

With no arguments, the `ls` command lists the objects in the current `RegistryPackage`. When a *pattern* or *URN* is provided, the command lists the objects in the current `RegistryPackage` whose names (in the current locale) or unique identifiers match *pattern* or *URN*.

Operands

pattern A pattern comprising literal characters and the special characters asterisk (*) (representing zero or more characters) and question mark (?) (representing one and only one character). You can specify more than one *pattern*.

URN A URN starting with `urn:`, for example, `urn:uuid:4a6741e7-4be1-4cfb-960a-e5520356c4fd`. You can specify more than one *URN*. The URN must be the unique identifier of the object, not the logical identifier.

Examples

The following command lists all the objects in the current `RegistryPackage`.

```
admin> ls
```

The following command lists all the objects whose name matches the pattern `urn:bird:poultry:chicken` or whose ID is `urn:bird:poultry:chicken`.

```
admin> ls urn:bird:poultry:chicken
```

The following command lists all the objects whose name matches the pattern `*bird*`. (It would also list the objects whose ID is `*bird*`, if `*bird*` were a valid ID.)

```
admin> ls *bird*
```

The following command lists all the objects whose name matches the pattern `*bird*` or whose name matches the pattern `urn:bird:poultry:chicken` or whose ID is `urn:bird:poultry:chicken`.

```
admin> ls *bird* urn:bird:poultry:chicken
```

pwd

Displays the path to the current RegistryPackage.

Synopsis

pwd

Description

The `pwd` command displays the path (or paths) to the current RegistryPackage using the best-matching names for the current locale. The command also displays the locale for the path.

Example

```
admin> pwd
(en_US) /registry/userData
```

quit

Exits the Admin Tool.

Synopsis

quit

Description

The `quit` command exits the Admin Tool.

Example

```
admin> quit
```

rm

Removes objects from a RegistryPackage.

Synopsis

```
rm [-d] [-r] {pattern | URN}...
```

Description

The `rm` command removes the member objects of the current `RegistryPackage` whose names (in the current locale) match the patterns specified by a *pattern* or *URN*.

When a matching `RegistryObject` is a member of multiple `RegistryPackage` objects, this command removes only the association between the current `RegistryPackage` and the object. The object is removed from the Registry only when the removal of the association leaves the object with no association with any other `RegistryObject`.

When a matching member object is itself a `RegistryPackage` that contains other objects, neither the object nor the association between the current `RegistryPackage` and the member `RegistryPackage` is removed unless either the `-r` or the `-d` option is specified.

When both the `-d` and `-r` options are specified, the `-d` option is applied recursively, so all objects that would be selected by `-r` (and their associations) are removed whether or not they have other associations.

Options

- d Removes the association between the current `RegistryPackage` and the specified `RegistryPackage`. Removes the specified `RegistryPackage` only if its only remaining associations are to its member objects. Member objects of the now-removed `RegistryPackage` that are not anchored by being the target of other `HasMember` associations are now accessible as members of the root of the Registry.
- r Removes the specified `RegistryPackage` object and all its descendant objects (except when an object has other associations).

Operands

pattern A pattern comprising literal characters and the special characters asterisk (*) (representing zero or more characters) and question mark (?) (representing one and only one character). You can specify more than one *pattern*.

URN A URN starting with `urn:`, for example, `urn:uuid:4a6741e7-4be1-4cfb-960a-e5520356c4fd`. You can specify more than one *URN*.

Examples

The following command removes all `RegistryPackage` objects that contain the string `"stat"` and all their descendants.

```
admin> rm -r *stat*
```

select

Executes an SQL `select` statement.

Synopsis

```
select [SQL]
```

Description

The `select` command selects and lists the objects that are specified by evaluating the entire command as an SQL query. If no argument is specified, the command lists any objects selected by a preceding `select` command or by the `-sqlselect` option.

Operand

SQL An SQL `select` statement (without the leading `select` because that is already present as the name of the command).

Examples

The following command lists all `ClassificationScheme` objects in the Registry:

```
admin> select s.* from ClassificationScheme s
```

set

Sets a property value.

Synopsis

```
set property value
```

Description

The `set` command sets the value of a property of the Admin Tool shell.

The tool supports the following properties and values.

```
set debug {true | on | yes | false | off | no}
```

Enables or disables output of debugging messages.

```
set editor string
```

Sets the command to use when the Admin Tool launches an interactive editor. The default value is `/bin/vi` on UNIX and Linux systems.

```
set verbose {true | on | yes | false | off | no}
```

Enables or disables output of more verbose messages when executing commands.

Operands

property One of the following properties: `debug`, `editor`, `verbose`.

value A supported value of the specified property. See the Description section for details.

Examples

The following command sets the editor to `/usr/bin/vi` instead of the default `/bin/vi`.

```
admin> set editor /usr/bin/vi
```

The following command turns on debugging.

```
admin> set debug true
```

The following command turns off verbose output.

```
admin> set verbose off
```

show

Displays a property value.

Synopsis

```
show [property]
```

Description

The `show` command displays the value of a property of the Admin Tool shell.

If no argument is specified, the command displays the values of all properties.

The command supports the following properties:

<code>debug</code>	Whether or not debugging output is enabled.
<code>editor</code>	The editor to use when the Admin Tool launches an interactive editor.
<code>exception</code>	The exception stack trace, if any, from the immediately preceding executed command.
<code>localdir</code>	The current directory on the local file system. Use the <code>lcd</code> command to set this property. See “ lcd ” on page 51 for details.
<code>locale</code>	The current locale.
<code>verbose</code>	Whether or not verbose output is enabled.

Operands

property The property whose current value is to be displayed. The properties `exception` and `locale` can be displayed, but you cannot use the `set` command to set them.

Example

The following command displays the exceptions from the previous command.

```
admin> show exception
```

users

Lists the current `User` objects.

Synopsis

```
users
```

Description

The `users` command lists the `User` objects currently in the Registry.

The output has the following format:

```
%index: URN lastname, firstname middlename
```

In the output, the *index* is a numeric value that you can use, including the percent sign (%), to refer to a user when you run the `chown` or `cp` command. The *lastname*, *firstname*, and *middlename* are the last, first, and middle names of the user.

Example

The following command displays the current users:

```
admin> users
%0: urn:freebxml:registry:predefinedusers:registryoperator Operator, Registry
%1: urn:freebxml:registry:predefinedusers:registryguest Guest, Registry
%2: urn:freebxml:registry:predefinedusers:farrukh Najmi, Farrukh Salahudin
%3: urn:freebxml:registry:predefinedusers:nikola Stojanovic, Nikola
%4: urn:uuid:799cc524-b7cd-4e51-8b34-d93b79ac52de User, Test
%5: urn:uuid:85428d8e-1bd5-473b-a8c8-b9d595f82728 Parker, Miles
```

Index

A

- add association command, 39-40
- add user command, 40-46
- AdhocQuery objects, adding to Web Console, 25-30
- Admin Console, Application Server, 19-20
- Admin Tool
 - command-line options, 35-36
 - enabling use of, 18
 - introduction, 33
 - starting, 34-36
 - stopping, 53
- administrators, creating, 24-25
- alias command-line option, 35
- Application Server Admin Console, 19-20
- Application Server domain
 - adding root certificates to truststore, 21
 - administering, 18-20
 - stopping and restarting, 20
- associations, adding to registry, 39-40

B

- batch mode, 34

C

- case sensitivity, 38
- cd command, 46-47
- chown command, 47-48
- command command-line option, 35

- command-line options, 35-36

- alias, 35
- command, 35
- create, 35
- debug, 35
- help, 35
- keypass, 36
- localdir, 36
- locale, 36
- registry, 36
- root, 36
- sqlselect, 36
- v, 36
- verbose, 36

- commands

- add association, 39-40
- add user, 40-46
- cd, 46-47
- chown, 47-48
- cp, 48-49
- echo, 49-50
- help, 50-51
- lcd, 51
- ls, 51-52
- pwd, 53
- quit, 53
- rm, 54-55
- select, 55-56
- set, 56-57
- show, 57-58
- users, 58

- configuring Service Registry, 15-23

- copying files and folders to Registry, 48-49

cp command, 48-49
-create command-line option, 35
creating administrators, 24-25
current directory, changing, 51

D

database for Service Registry, backing up and restoring, 32
-debug command-line option, 35
debug property
 displaying value, 57-58
 setting, 56-57
deleting objects from RegistryPackage, 54-55
directory, changing, 51
displaying, property values, 57-58

E

echo command, 49-50
editor property
 displaying value, 57-58
 setting, 56-57
exception property, displaying value, 57-58
exceptions, displaying, 37
exiting the Admin Tool, 53
external web sites, allowing access to, 23
ExternalLink objects, allowing validation of URLs, 23

F

file system, local
 base directory, 36
 changing current directory, 51
files and folders, copying to Registry, 48-49

G

Glossary, link to, 10

H

help command, 50-51
-help command-line option, 35

I

installation properties, 16-17
interactive mode, 34-35

K

-keypass command-line option, 36

L

lcd command, 51
-localdir command-line option, 36
-locale command-line option, 36
locale property, displaying value, 57-58
locales, effect on specifying names, 38
ls command, 51-52

P

ports, Service Registry default, 19
predefined queries, adding to Web Console, 25-30
property values
 displaying, 57-58
 setting, 56-57
proxy host and port, setting, 23
pwd command, 53

Q

quit command, 53

R

-registry command-line option, 36
registry domain
 adding root certificates to truststore, 21

- registry domain (Continued)
 - administering, 18-20
 - stopping and restarting, 20
- registry objects
 - changing owner, 47-48
 - identifying, 37
 - listing, 51-52
- RegistryPackage location, changing, 46-47
- RegistryPackage objects
 - creating, 35
 - displaying path to, 53
 - listing contents, 51-52
 - removing member objects, 54-55
- removing objects from RegistryPackage, 54-55
- rm command, 54-55
- root certificates, adding to registry domain
 - truststore, 21
- root command-line option, 36

S

- select command, 55-56
- ServiceBinding objects, allowing validation of
 - URLs, 23
- set command, 56-57
- setting property values, 56-57
- show command, 57-58
- SQL select statements, executing, 36
- SQL statements, executing, 55-56
- sqlselect command-line option, 36
- starting the Admin Tool, 34-36
- stopping the Admin Tool, 53

U

- user name, specifying on command line, 35
- users
 - adding to registry, 40-46
 - listing, 58
- users command, 58

V

- v command-line option, 36
- verbose command-line option, 36

- verbose property
 - displaying value, 57-58
 - setting, 56-57

W

- Web Console, adding predefined queries, 25-30
- wildcards, using, 37

