

Oracle® Secure Enterprise Search

Administration API Guide

11g Release 1 (11.1.2.2)

E21606-01

April 2011

Oracle Secure Enterprise Search Administration API Guide, 11g Release 1 (11.1.2.2)

E21606-01

Copyright © 2009, 2011, Oracle and/or its affiliates. All rights reserved.

Primary Author: Donna Carver

Contributor: Greg Brunet, Belinda Leung, Yan Zhao

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xiii
Audience	xiii
Documentation Accessibility	xiii
Related Documents	xiii
Conventions	xiv
1 Using the Administration API	
Introduction to the Administration API	1-1
Command Interface	1-1
Opening an Interactive Session	1-2
Issuing Individual Commands	1-2
Issuing Commands to Remote Oracle SES Instances	1-3
Using the Help Command	1-3
Command Syntax	1-4
Web Services Interface	1-5
Connecting to the Web Services Endpoint	1-5
Providing Credentials	1-5
Using the Web Services Java Client	1-5
Java Libraries	1-5
Creating a Stateful Web Services Client	1-6
Creating a Stateless Web Services Client	1-6
Creating an Administrative Object	1-7
Object Types	1-7
Universal Objects	1-7
Creatable Types	1-8
Object Properties	1-9
XML Documents	1-9
Sample XML Document of an Administration Object	1-9
Editing XML Files	1-9
State Properties	1-10
Operations	1-10
Command Syntax	1-10
Using the Message Logs	1-11
Log Files	1-11
Configuration File	1-11

Managing Universal Objects	1-12
Managing Creatable Objects.....	1-12
Managing Object State.....	1-12
Status Codes for Bulk Operations.....	1-13

2 Administration Object Types

Alphabetic List of Administration Object Types	2-1
Document Support	2-2
Globalization Support	2-3
Product Languages	2-3
Crawlable Documents	2-4
Providing Translations of Object Names.....	2-6
Encryption	2-8
XML Description of State Properties	2-8
Partitioning for Parallel Query	2-10
Disk Space Management: Quotas and Alerts	2-12
Search Interface Customization: Skin Bundles	2-13
Support Bundles.....	2-14
FreeMarker Templates.....	2-14
Asset Files.....	2-14
JavaScript Libraries	2-15
Template Library.....	2-15
Configuration Settings.....	2-15
Assembling the Skin Bundle Files	2-16
Creating a skinBundle Object.....	2-16
Using a Skin Bundle to Render the Search Application User Interface	2-17
Skin Bundle Example.....	2-17
Changes to the Example Results Page	2-17
Changes to the Example Footer	2-18
Creating the Example Directory Structure.....	2-19
Customizing results.ftl	2-19
Customizing inc_logo_querybox.ftl.....	2-20
Customizing inc_footer.ftl.....	2-21
Creating the Example Skin Bundle File List	2-21
Creating an XML Description of the Example Skin Bundle	2-21
Creating the Example skinBundle Object.....	2-22
Using the Example Skin Bundle to Render the Search Application.....	2-22
alert	2-23
altWord	2-26
clustering	2-28
clusterTree	2-30
crawlerSettings	2-34
identityPlugin	2-38
index	2-42
indexOptimizer	2-44
partitionConfig	2-47
proxyLogin	2-49

queryConfig.....	2-51
resultList.....	2-57
schedule.....	2-59
searchAttr.....	2-63
skinBundle.....	2-66
source.....	2-68
sourceGroup.....	2-93
sourceType.....	2-95
spaceCalculator.....	2-99
storageArea.....	2-102
suggLink.....	2-105
task.....	2-107
thesaurus.....	2-108

3 searchadmin Commands

Alphabetic List of searchadmin Commands.....	3-1
Entering Special Characters.....	3-3
activate clustering.....	3-4
activate clusterTree.....	3-5
activate identityPlugin.....	3-6
activate indexOptimizer.....	3-7
activate resultList.....	3-8
activate schedule.....	3-9
activate skinBundle.....	3-10
activate spaceCalculator.....	3-11
create altWord.....	3-12
create clusterTree.....	3-13
create identityPlugin.....	3-14
create proxyLogin.....	3-15
create schedule.....	3-16
create searchAttr.....	3-17
create skinBundle.....	3-18
create source.....	3-19
create sourceGroup.....	3-20
create sourceType.....	3-21
create storageArea.....	3-22
create suggLink.....	3-23
create thesaurus.....	3-24
createAll altWord.....	3-25
createAll clusterTree.....	3-26
createAll identityPlugin.....	3-27
createAll proxyLogin.....	3-28
createAll schedule.....	3-29
createAll searchAttr.....	3-30
createAll source.....	3-31
createAll sourceGroup.....	3-32
createAll sourceType.....	3-33

createAll storageArea.....	3-34
createAll suggLink	3-35
deactivate clustering	3-36
deactivate clusterTree	3-37
deactivate identityPlugin.....	3-38
deactivate indexOptimizer	3-39
deactivate resultList	3-40
deactivate schedule	3-41
deactivate skinBundle	3-42
deactivate spaceCalculator.....	3-43
delete <i>creatable_type</i>	3-44
deleteAll <i>creatable_type</i>	3-45
deleteList <i>creatable_type</i>	3-47
export <i>creatable_type</i>	3-49
export <i>universal_type</i>	3-51
exportAll <i>creatable_type</i>	3-53
exportList <i>creatable_type</i>	3-55
getAllObjectKeys	3-57
getAPIVersion	3-59
getAllStates clusterTree.....	3-60
getAllStates identityPlugin	3-62
getAllStates schedule	3-64
getAllStates skinBundle	3-66
getState clustering	3-68
getState clusterTree	3-69
getState identityPlugin.....	3-71
getState index	3-73
getState indexOptimizer	3-74
getState resultList.....	3-76
getState schedule	3-77
getState skinBundle	3-79
getState spaceCalculator	3-81
getState task.....	3-82
getStateList clusterTree	3-84
getStateList identityPlugin.....	3-86
getStateList schedule	3-88
getStateList skinBundle	3-90
start indexOptimizer.....	3-91
start schedule.....	3-92
start spaceCalculator	3-93
start task	3-94
stop schedule	3-95
update alert	3-96
update altWord.....	3-97
update clustering	3-98
update clusterTree	3-99
update crawlerSettings.....	3-100

update index	3-101
update indexOptimizer	3-102
update partitionConfig.....	3-103
update proxyLogin	3-104
update queryConfig.....	3-105
update resultList.....	3-106
update schedule	3-107
update searchAttr	3-108
update skinBundle.....	3-109
update source	3-110
update sourceGroup	3-111
update sourceType	3-112
update spaceCalculator	3-113
update storageArea	3-114
update suggLink.....	3-115
update thesaurus	3-116
updateAll alert	3-117
updateAll altWord.....	3-118
updateAll clusterTree	3-119
updateAll proxyLogin	3-120
updateAll schedule	3-121
updateAll searchAttr.....	3-122
updateAll source.....	3-123
updateAll sourceGroup.....	3-124
updateAll sourceType.....	3-125
updateAll storageArea.....	3-126
updateAll suggLink	3-127

4 Web Service Operations

Alphabetic List of Web Service Operations	4-1
activate	4-3
create	4-6
createAll	4-9
deactivate	4-13
delete.....	4-15
deleteAll	4-17
deleteList.....	4-20
export	4-24
exportAll.....	4-27
exportList	4-30
getAllObjectKeys	4-33
getAPIVersion	4-35
getAllStates.....	4-37
getState	4-40
getStateList	4-43
login	4-46
logout.....	4-47

start	4-48
stop	4-50
update	4-52
updateAll	4-56
Java Source Code Example	A-1
Shell Script Example	A-5
Invalid Input Errors	B-1
Creatable Administration Object Errors	B-10
Invalid State Errors	B-11
Administration API Run-Time Errors	B-12
Dependent Object Errors	B-13

List of Examples

1-1	Issuing Commands at the SES Prompt	1-2
1-2	Issuing Commands at the Operating System Prompt	1-2
2-1	Federated Source Description	2-71
2-2	File Source Description	2-76
2-3	User-Defined Source Description	2-80
2-4	Web Source Description	2-89

List of Tables

1-1	Help Syntax.....	1-4
1-2	Universal Objects	1-8
1-3	Creatable Object Types.....	1-8
1-4	Administrative Objects With State Properties.....	1-10
1-5	Logging Properties.....	1-11
1-6	Operations on Universal Objects	1-12
1-7	Operations on Creatable Objects	1-12
1-8	Operations on Objects With State Properties.....	1-12
1-9	Status Codes.....	1-13
2-1	Document Formats	2-2
2-2	Product Languages	2-3
2-3	Crawlable Languages	2-4
2-4	Crawlable Character Sets.....	2-5
2-5	Query Language Codes.....	2-7
2-6	Differences Between the Default Skin and the Example Skin.....	2-17

Preface

The *Oracle Secure Enterprise Search Administration API Guide* documents two interfaces to the Administration API: a command-line interface and a Web services interface. It also introduces the Web services Java client. The Administration API supports the same features as the Oracle SES Administration GUI, and some new features that are currently not supported by the GUI.

Audience

This document is intended for anyone using the Oracle SES Administration API:

- **Administrators** of Oracle Secure Enterprise Search may find a command-line interface to be easier for routine tasks than a graphical user interface. Administrators of large installations, who make the same changes across many instances of Oracle SES, will find the command-line interface to be particularly useful.
- **Java developers** can create custom administrative tools using the Web services Java client.
- **Web services developers** can create custom administrative tools using any technology that supports Simple Object Access Protocol (SOAP), such as Java and Microsoft ASP.NET.
- **Web designers** can create custom skins for the default search interface using any of these interfaces.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/us/corporate/accessibility/index.html>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

For more information about Oracle Secure Enterprise Search, refer to the following resources:

- *Oracle Secure Enterprise Search Administrator's Guide*
Explains how to administer Oracle Secure Enterprise Search instances, including how to set up a variety of information sources, crawl and index those sources, and customize the search results.
- *Oracle Secure Enterprise Search Release Notes*
Provides version information and identifies known issues.
- *Oracle Secure Enterprise Search Installation Guides*
Discuss installation requirements and tips, and provides information on how to get started using Oracle Secure Enterprise Search.
- *Oracle Secure Enterprise Search Java API Reference*
Describes the classes and methods in the Oracle SES Java APIs.

Up-to-date Release Notes are posted on Oracle Technology Network (OTN). You must register online before using OTN. Registration is free and can be done at this location:

<http://www.oracle.com/technology>

If you have a user name and password for OTN, then you can go directly to the documentation section of OTN at this location:

<http://www.oracle.com/technology/documentation>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

This reference presents syntax in a simple variant of Backus-Naur Form (BNF) that includes the following symbols and conventions.

Symbol or Convention	Meaning
[]	Brackets enclose optional items.
{ }	Braces enclose a choice of items of which only one is required.
	A vertical bar separates alternatives.
...	Ellipses indicate that the preceding syntactic element can be repeated.
/	A slash separates levels of a directory path. On Windows, use a backslash (\) in place of a slash (/).
delimiters	Delimiters other than brackets, braces, vertical bars, and ellipses must be entered as shown.

Symbol or Convention	Meaning
<i>italics</i>	Words appearing in italics are placeholders for which you must substitute a name or a value. Words that are not in italics are keywords and must be entered as shown.
<i>ORACLE_BASE</i>	<i>ORACLE_BASE</i> is the root of the directory structure, where Oracle SES software and its supporting infrastructure software are installed. This directory path typically is stored in an environment variable. On Linux and UNIX systems, you can reference the variable as \$ORACLE_BASE. On Windows, the equivalent is %ORACLE_BASE%.
<i>ORACLE_HOME</i>	<i>ORACLE_HOME</i> is the <i>ORACLE_BASE</i> /seshome directory. This directory path typically is stored in an environment variable. On Linux and UNIX systems, you can reference the variable as \$ORACLE_HOME. On Windows, the equivalent is %ORACLE_HOME%.

Using the Administration API

This chapter explains basic concepts and use of the Administration API. It contains the following topics:

- [Introduction to the Administration API](#)
- [Command Interface](#)
- [Web Services Interface](#)
- [Object Types](#)
- [Object Properties](#)
- [Operations](#)

Introduction to the Administration API

The Oracle SES Administration API supports management of large-scale deployments. It provides a command-line interface and a Web services interface to the same administrative tasks performed using the Oracle SES Administration GUI.

The following are the building blocks of the Administration API:

- **Administrative Objects:** An administrative object (or simply an object) models a feature in Oracle SES that can be managed directly through the API.

An object is either creatable or universal. You can create multiple instances of a creatable object, such as a source or a schedule. You can configure, but not create, a universal object, such as the crawler settings. The administrative objects are described in [Chapter 2, "Administration Object Types."](#)

- **Operations:** Operations perform an action on one or more objects, such as creating, deleting, starting, or stopping them. If an operation fails, then all changes are rolled back. The command-line operations are described in [Chapter 3, "searchadmin Commands,"](#) and the Web services operations are described in [Chapter 4, "Web Service Operations."](#)

Command Interface

The `searchadmin` command provides a command-line interface to the Administration API. You can open an interactive session, or you can issue individual commands to the operating system.

You must supply the administrator password each time you issue the `searchadmin` command. You can include the password in the command or wait for the prompt to enter it.

Opening an Interactive Session

To open an interactive session, enter the `searchadmin` command at the operating system prompt. After providing the administrator password, you get the SES prompt. You can start entering commands to the Oracle SES Administration API. This type of connection is called **session mode**.

[Example 1-1](#) shows a brief interactive session, which ends with a `quit` command.

Example 1-1 Issuing Commands at the SES Prompt

```
$ searchadmin

Search Admin Command Line - Release 11.1.2.2.0

Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.

Password:

SES>getAPIVersion

11.1.2.2.0

SES>export index

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:index>
    <search:indexingBatchSize>250</search:indexingBatchSize>
    <search:indexingMemorySize>275</search:indexingMemorySize>
  </search:index>
</search:config>

SES>quit
$
```

Issuing Individual Commands

You can issue commands individually at the operating system prompt. You can also use this command format to create scripts and batch files, thus simplifying routine tasks. This type of connection is called **single job mode**.

[Example 1-2](#) shows two commands entered at the operating system prompt. One includes the password in the command, and the other responds to the Password prompt. In a script or batch file, include the password in each command.

Example 1-2 Issuing Commands at the Operating System Prompt

```
$ searchadmin getAPIVersion

Search Admin Command Line - Release 11.1.2.2.0

Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.

Password:

11.1.2.2.0

$ searchadmin --PASSWORD password export index
```

```
Search Admin Command Line - Release 11.1.2.2.0
```

```
Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:index>
    <search:indexingBatchSize>250</search:indexingBatchSize>
    <search:indexingMemorySize>275</search:indexingMemorySize>
  </search:index>
</search:config>

$
```

Issuing Commands to Remote Oracle SES Instances

The `searchadmin` command connects to the local search instance defined by Oracle home. However, you can issue commands to one or more remote Oracle SES instances by supplying the connection information in the `searchadmin` command.

To connect to a single remote instance, use the `--CONNECTION` option with an HTTP connection string, like the one shown here. You can obtain the port number from the Oracle SES properties file at `ORACLE_HOME/search/webapp/config/search.properties`.

```
$ searchadmin --CONNECTION=http://myhost:7777/search/api/admin/AdminService
```

To connect to multiple instances, create a file with the connection information, then reference the file using the `--CONNECTION_LIST` option. Each command is executed across all instances.

This is the file format:

```
--USERNAME=eqsys --CONNECTION=http://myhost:7777/search/api/admin/AdminService
```

This command opens connections using a file named `remotehosts.lst`:

```
$ searchadmin --CONNECTION_LIST=remotehosts.lst
```

Using the Help Command

The `Help` command provides the basic command syntax and descriptions of administrative objects. You can enter the help command in an interactive session or as an individual command, as shown here:

```
$ searchadmin help
```

```
Search Admin Command Line - Release 11.1.2.2.0
```

```
Copyright (c) 2006, 2010, Oracle and/or its affiliates. All rights reserved.
```

```
Usage: searchadmin [<connectionArgs>] [<operation>] [<operationArgs>]
```

```
.
.
.
```

[Table 1–1](#) describes the `searchadmin` help commands.

Table 1–1 Help Syntax

Command	Description
help	Provides the <code>searchadmin</code> command syntax, and lists the administrative object types and operations.
help <i>object_type</i>	Describes <i>object_type</i> .
help <i>operation</i>	Provides the command syntax for <i>operation</i> .
help statusCodes	Describes the status codes that are displayed after an operation is performed on multiple objects by <code>createAll</code> , <code>deleteAll</code> , <code>deleteList</code> , or <code>updateAll</code> .
help duplicateMethod	Describes the settings of the <code>--DUPE_METHOD</code> parameter in the <code>createAll</code> command.
help updateMethod	Describes the settings of the <code>--UPDATE_METHOD</code> parameter in the <code>update</code> and <code>updateAll</code> commands.
help notFoundMethod	Describes the settings of the <code>--NOT_FOUND_METHOD</code> parameter in the <code>updateAll</code> command.

Command Syntax

The following is a summary of the `searchadmin` command syntax.

```
searchadmin [--USERNAME=admin] [--PASSWORD=password] [--CONNECTION=connection] |
--CONNECTION_LIST=filename}] [operation]
```

or

```
searchadmin [-u admin] [-p password] [--c connection | -m filename}] [operation]
```

admin

Administrator user name, which is `eqsys` by default.

password

Password for the administrator. You are prompted for the password if you omit it from the command. If you are connecting to multiple instances, the password is used for authentication on all of them; omit this argument if they do not use the same password.

connection

Connection to a remote Oracle SES instance. The connection is to the local instance identified by Oracle home. You can connect using HTTP, using the following syntax:

```
http://host:port/search/api/admin/AdminService
```

The port number for HTTP duplicates the port for the Administration GUI.

filename

The name of the connection file. The path can be absolute or relative to the current directory.

operation

An administrative operation, as described in [Chapter 3, "searchadmin Commands."](#) Omit this argument to open an interactive session. Like the `searchadmin` command, the operations have a short form and a long form for many arguments.

Web Services Interface

The Oracle SES Web services interface enables you to create client applications easily in a variety of technologies that support Simple Object Access Protocol (SOAP), such as Java and Microsoft ASP.NET. Moreover, two Web services clients are available out-of-the-box:

- A Java client enables you to develop Java programs that use the Administration API.
- A WebLogic test client enables you to test individual SOAP requests to the Administration API, thus facilitating development of new client applications.

Connecting to the Web Services Endpoint

The endpoint for the Web service has the following URL:

```
http://host:port/search/api/admin/AdminService
```

The host name and port number are the same ones that you use to connect to the Administration GUI.

The endpoint page provides this information:

- **Service Name:** {http://search.oracle.com/Admin}AdminService
- **Port Name:** {http://search.oracle.com/Admin}Admin
- **Address:** http://host:port/search/api/admin/AdminService
- **WSDL:** http://host:port/search/api/admin/AdminService?wsdl
- **Implementation class:**
oracle.search.admin.api.ws.OracleSearchAdminImpl

Providing Credentials

Like the command-line interface, the Web services interface has two modes of operation:

- **Stateful:** The administrative user name and password are provided at the beginning and retained for all subsequent operations, while the client maintains the HTTP session. Use this mode when executing a sequence of operations, and thus achieve the best performance. You can provide credentials in any operation. Your application does not have to retain the password.
- **Stateless:** The administrative user name and password are provided for each operation. Use this mode to avoid maintaining open HTTP connections when, for example, use of the application is sporadic.

Using the Web Services Java Client

Using the Java client, you can develop your own programs for managing Oracle SES instances. The *Oracle Secure Enterprise Search Java API Reference* describes the classes and methods.

See Also: [Appendix A, "Java Example"](#)

Java Libraries

This is the Java Web services client JAR file:

```
ORACLE_HOME/search/lib/search_adminapi_wsclient.jar
```

It requires the following version or JRockit or later, or any JDK 6 that contains a JAX-WS 2.1 implementation. Depending on the platform, you can find these jar files in the following locations:

`ORACLE_BASE/jrockit_160_14_R27.6.5-32/jre/lib/rt.jar`

`ORACLE_BASE/jdk6/jre/lib/rt.jar`

Creating a Stateful Web Services Client

The following Java code fragment creates a stateful client:

```
//Initialize and return a stateful admin web service client

private static AdminPortType getStatefulWebServiceClient(
    String webServiceURL,
    String userName,
    String password) throws Exception
{
    AdminService adminService = new AdminService(
        new URL( webServiceURL ),
        new QName(
            "http://search.oracle.com/Admin",
            "AdminService"
        )
    );

    AdminPortType adminPort = adminService.getAdmin();

    // Tell client proxy to maintain HTTP session for stateful behavior
    ((BindingProvider)adminPort).getRequestContext().put(
        BindingProvider.SESSION_MAINTAIN_PROPERTY, true
    );

    // Create credentials argument
    Credentials credentials = new Credentials();
    credentials.setUsername( userName );
    credentials.setPassword( password );

    adminPort.login( credentials, "en" );

    return adminPort;
}
```

Creating a Stateless Web Services Client

This Java code fragment creates a stateless client. It does not provide credentials.

```
// Initialize and return a stateless admin web service client
private static AdminPortType getStatelessWebServiceClient(
    String webServiceURL) throws Exception
{
    AdminService adminService = new AdminService(
        new URL( webServiceURL ),
        new QName(
            "http://search.oracle.com/Admin",
            "AdminService"
        )
    );
    return adminService.getAdmin()
}
```

You can use the stateless client by providing credentials for each operation.

```
// Get stateless web service client
adminPort = getStatelessWebServiceClient( webServiceURL );

// Create Credentials object for operation
Credentials credentials = new Credentials();
credentials.setUserName( userName );
credentials.setPassword( password );
```

Creating an Administrative Object

This Java code fragment creates a Web source named `example`.

```
String webSourceURL = "http://www.example.com";

String webSourceXML =
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
"<search:config productVersion=\"11.1.2.2.0\"" +
xmlns:search=\"http://xmlns.oracle.com/search\">" +
"  <search:sources>" +
"    <search:webSource>" +
"      <search:name>example</search:name>" +
"      <search:startingUrls>" +
"        <search:startingUrl>" +
"          <search:url>" + webSourceURL + "</search:url>" +
"        </search:startingUrl>" +
"      </search:startingUrls>" +
"    </search:webSource>" +
"  </search:sources>" +
"</search:config>";

adminPort.createAll(
    "source",
    webSourceXML,
    "password",
    credentials,
    null,
    null,
    "en"
);
```

Object Types

The Administration API enables you to perform a variety of operations on the administrative objects. These objects fall into two basic categories: universal and creatable.

Universal Objects

Oracle SES has one instance of each universal object out of the box. You can change the default settings, but you cannot create a new instance or delete the existing one. In the Administration GUI, you can edit the settings of universal objects on the Global Settings pages.

[Table 1–2](#) describes the universal object types.

Table 1–2 Universal Objects

Object Type	Description
<code>clustering</code>	Clustering configuration
<code>crawlerSettings</code>	Crawler configuration
<code>index</code>	Indexing parameters
<code>indexOptimizer</code>	Index optimization
<code>partitionConfig</code>	Partition configuration
<code>queryConfig</code>	Query configuration
<code>resultList</code>	Search result list configuration
<code>spaceCalculator</code>	Space calculator

Creatable Types

Oracle SES may have multiple instances of a creatable type out of the box, or it may have none. You can create new instances and, for most types, modify existing ones. In the Administration GUI, you can create and edit most of these objects on the Home and Search pages.

An object key uniquely identifies a particular instance of a creatable type. The key can be a single value, such as the name of a source, or a composite value, such as the jar file name and class of an identity plug-in. If an object key contains spaces, then all references must be enclosed in quotes in the command-line interface:

```
--NAME=this_DataSource
--NAME="This Data Source"
```

Table 1–3 describes the creatable objects.

Table 1–3 Creatable Object Types

Object Type	Description
<code>alert</code>	Alert
<code>altWord</code>	Alternate words
<code>clusterTree</code>	Cluster trees
<code>identityPlugin</code>	Identity plug-ins
<code>proxyLogin</code>	Proxy log-ins
<code>schedule</code>	Schedules
<code>searchAttr</code>	Search attributes
<code>skinBundle</code>	Skin bundle
<code>source</code>	Sources
<code>sourceGroup</code>	Source groups
<code>sourceType</code>	Source types
<code>storageArea</code>	Storage areas
<code>suggLink</code>	Suggested links
<code>task</code>	Task
<code>thesaurus</code>	Thesaurus

Object Properties

All object types have properties that configure the object to operate in a particular way. An XML document describes these properties. When you create or modify an object, you submit an XML document to Oracle SES that describes the object and sets the values of its properties. When you query Oracle SES for a description of an object, it returns the information as an XML document.

XML Documents

Although you can develop XML descriptions of administration objects from the start, an easier method is to let Oracle SES do the work for you. For universal objects, you can export the description to a file for editing. For creatable objects, you can use the Administration GUI to develop an object and then export the description to a file. You can then edit the file or copy it to create similar objects. This method is particularly useful when creating or updating complex object types, such as sources.

Sample XML Document of an Administration Object

The following is a very simple example of an XML document generated by Oracle SES. It describes the universal index object:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:index>
    <search:indexingBatchSize>250</search:indexingBatchSize>
    <search:indexingMemorySize>275</search:indexingMemorySize>
  </search:index>
</search:config>
```

<?xml version="1.0" encoding="UTF-8"?>

Identifies the file as an XML document. The encoding of exported XML is set to UTF-8, but imported documents can have other encoding.

<search:config>

Contains all Oracle SES object configuration elements. This is the root element of the XML document.

<search:index>

Contains the configuration settings for the universal index element. You can describe administration objects in individual XML files or all objects in one XML file.

[Chapter 2, "Administration Object Types,"](#) discusses the XML description of each object type.

Editing XML Files

The XML complies with the standard conventions for XML documents. You should have a working knowledge of XML before using the administration APIs.

XML is a text-based markup language, so you can use any text editor to create and edit XML files. However, an ordinary text editor cannot detect when the document is correctly formed and, more importantly, when it contains syntax errors. For that, you should use an XML editor. You can choose from professional and freeware versions that are available for download on the Web.

An XML editor can also check an XML document against the XML Schema Definition (XSD). Oracle SES has two files for the XSD:

- **Config.xsd:** Contains the object definitions.
- **State.xsd:** Contains the schema for the `getState`, `getStateList`, and `getAllStates` operations.

Both files are in `ORACLE_HOME/search/xsd/11_1_2_0_0/adminapi`.

State Properties

Some object types have state properties that provide information about the current state of the object. For example, a cluster tree has a `status` property that reports whether it is enabled or disabled. Both universal and creatable object types can have state properties. [Chapter 2, "Administration Object Types,"](#) lists the state properties for each object type.

Oracle SES returns the current state of an object in the form of an XML document. See ["XML Description of State Properties"](#) on page 2-8.

Table 1–4 Administrative Objects With State Properties

Object Type	State Properties
<code>clustering</code>	<code>status</code>
<code>clusterTree</code>	<code>status</code>
<code>identityPlugin</code>	<code>status</code>
<code>index</code>	<code>estimatedFragmentation</code>
<code>indexOptimizer</code>	<code>endTime</code> , <code>startTime</code> , <code>status</code>
<code>resultList</code>	<code>status</code>
<code>schedule</code>	<code>lastCrawled</code> , <code>logFilePath</code> , <code>nextCrawl</code> , <code>scheduleError</code> , <code>status</code>
<code>skinBundle</code>	<code>status</code>
<code>spaceCalculator</code>	<code>endTime</code> , <code>error</code> , <code>startTime</code> , <code>status</code>
<code>task</code>	<code>disabledSchedules</code> , <code>endTime</code> , <code>error</code> , <code>startTime</code> , <code>status</code> , <code>stoppedSchedules</code>

Operations

You execute various operations to manage the administration objects. [Chapter 3, "searchadmin Commands,"](#) provides the syntax of these operations and command examples. The same operations are available as XML documents in the Web services interface, described in [Chapter 4, "Web Service Operations."](#)

Command Syntax

The commands have this general syntax for universal object types:

```
operation object_type [parameters] [options]
```

The syntax for creatable types includes the object key for operations performed on a single object:

```
operation object_type [object_key] [parameters] [options]
```

For example, the following command activates clustering. The clustering object is a universal object type, and the command consists only of the required *operation* and *object_type*:

```
activate clustering
```

The next command updates the clustering configuration. This command requires the path to the input XML file with the new configuration settings and an update method:

```
update clustering --INPUT_FILE=clustering.xml --UPDATE_METHOD=overwrite
```

Most parameters and some common options have a shortcut notation. The previous update command can also be expressed with this syntax:

```
update clustering -i clustering.xml -a overwrite
```

Sources are creatable, so the command to change the configuration of a source requires the object key. For sources, the object key is the name. The following command also includes the `INPUT_FILE` parameter.

```
update source --NAME="Doc Library" --UPDATE_METHOD=overwrite --INPUT_
FILE=sources.xml
```

The shortcut notation looks like this:

```
update source -n "Doc Library" -a overwrite -i sources.xml
```

Using the Message Logs

The message logs can help you debug problems executing an operation.

Log Files

The `searchadmin` command logs messages in

`ORACLE_HOME/search/log/searchadmin/searchadmin.log`.

The WebLogic server for Web services logs messages in

`ORACLE_HOME/search/base_
domain/servers/AdminServer/logs/AdminServer.out`.

Configuration File

The Administration API uses Apache log4j logging services. The configuration file sets the logging level and other properties. To change the settings, edit

`ORACLE_HOME/search/config/searchadmin/log4j.properties`.

Table 1–5 describes the properties.

Table 1–5 Logging Properties

Property	Description
<code>rootLogger</code>	Sets the level of detail: <ul style="list-style-type: none"> ■ <code>DEBUG</code>: Tracks the status of operations throughout execution and logs details about common errors, such as invalid input. ■ <code>INFO</code>: Generates important status messages, such as a warning that a federated source cannot be contacted. (Default)
<code>File</code>	Path to the log file, initially set to <code>ORACLE_HOME/search/log/searchadmin/searchadmin.log</code> .
<code>Append</code>	Set to append messages to the file (default), or <code>false</code> to overwrite it.

Table 1–5 (Cont.) Logging Properties

Property	Description
MaxFileSize	Maximum size of the log file, initially set to 1MB. It starts over when it reaches that size.
MaxBackupIndex	Number of backup files. Initially set to 1.
layout	Describes the layout of the messages.

See Also: Documentation for Apache Logging Services log4j 1.2:

<http://logging.apache.org>

Managing Universal Objects

Use these operations to manage universal administration objects:

Table 1–6 Operations on Universal Objects

Operation	Description
export	Returns the XML description of an object.
update	Sets the parameters of an object from an XML file.

Managing Creatable Objects

Use these operations to manage creatable administration objects:

Table 1–7 Operations on Creatable Objects

Operation	Description
create	Creates an object from an XML file.
createAll	Creates all the objects of a particular type from an XML file.
delete	Deletes a single object.
deleteAll	Deletes all objects of a particular type.
deleteList	Deletes a list of objects of a particular type.
export	Returns the XML description of an object.
exportAll	Returns the XML descriptions of all objects of a particular type.
exportList	Returns the XML descriptions of a list of objects of a particular type.
update	Sets the parameters of an object from an XML file.
updateAll	Sets the parameters of all objects of a particular type from an XML file.

Managing Object State

Use these operations to manage both universal and creatable administration objects with state properties.

Table 1–8 Operations on Objects With State Properties

Operation	Description
activate	Enables an object. Objects that can be enabled have a status state property.

Table 1–8 (Cont.) Operations on Objects With State Properties

Operation	Description
deactivate	Disables an object. Objects that can be disabled have a <code>status</code> state property.
getAllStates	Returns the current state of all objects of a particular type as an XML document.
getState	Returns the state of an object as an XML document.
getStateList	Returns the state of a list of objects of a particular type as an XML document.
start	Initiates the starting process of an object.
stop	Initiates the stopping process of an object.

Status Codes for Bulk Operations

Operations that involve multiple objects, such as `createAll`, `deleteAll` and `deleteList`, return status codes for each processed object, as described in [Table 1–9](#).

Table 1–9 Status Codes

Status Code	Description
CREATE_NOT_SUPPORTED	The object cannot be created.
CREATE_SUCCEEDED	The object was successfully created.
DELETE_NOT_SUPPORTED	The object cannot be deleted.
DELETE_SUCCEEDED	The object was successfully deleted.
DUPLICATE_IGNORED	The object already existed. The operation ignored the pre-existing object.
DUPLICATE_OVERWRITTEN	The object already existed. The operation overwrote the pre-existing object.
INVALID_STATE_IGNORED	The object was ignored because it was in an invalid state.
NOT_FOUND_CREATED	The object did not exist. The operation created the object.
NOT_FOUND_IGNORED	The object did not exist. The operation ignored the object.
UPDATE_NOT_SUPPORTED	The object cannot be updated.
UPDATE_SUCCEEDED	The object was successfully updated.

Administration Object Types

This chapter describes the object types in the Oracle SES Administration API. It contains these topics:

- [Alphabetic List of Administration Object Types](#)
- [Document Support](#)
- [Globalization Support](#)
- [Encryption](#)
- [XML Description of State Properties](#)
- [Partitioning for Parallel Query](#)
- [Disk Space Management: Quotas and Alerts](#)
- [Search Interface Customization: Skin Bundles](#)

Alphabetic List of Administration Object Types

ACIPQRST

A

[alert](#)
[altWord](#)

C

[clustering](#)
[clusterTree](#)
[crawlerSettings](#)

I

[identityPlugin](#)
[index](#)
[indexOptimizer](#)

P

[partitionConfig](#)
[proxyLogin](#)

Q

[queryConfig](#)

R[resultList](#)**S**[schedule](#)[searchAttr](#)[skinBundle](#)[source](#)[sourceGroup](#)[sourceType](#)[spaceCalculator](#)[storageArea](#)[suggLink](#)**T**[task](#)[thesaurus](#)

Document Support

Table 2–1 identifies the document formats supported by Oracle SES.

Table 2–1 Document Formats

Document Format	MIME Type
Adobe Framemaker Document	application/x-frameset
Adobe Framemaker Interchange Format (MIF) Document	application/vnd.mif
Corel Presentations Document	application/vnd.corel-presentations
DICOM Image	application/dicom
DocuShare Ichitaro Document	application/x-js-taro
GIF Image	image/gif
GNU ZIP Archive	application/x-gzip
Haansoft HWP Document	application/x-hwp
HTML	text/html
JPEG 2000 Image	image/jp2
JPEG Image	image/jpeg
Lotus 1-2-3 Document	application/x-lotus123 (also represents application/vnd.lotus-1-2-3)
Lotus AMI Pro Document	application/x-ami
Lotus Freelance Document	application/x-freelance (also represents application/vnd.lotus-freelance)
Lotus Word Pro Document	application/vnd.lotus-wordpro
LZH Archive	application/x-lzh-compressed
Microsoft Excel Document	application/x-msexcel (also represents application/vnd.ms-excel and application/ms-excel)
Microsoft Office Project	application/vnd.ms-project

Table 2–1 (Cont.) Document Formats

Document Format	MIME Type
Microsoft PowerPoint Document	application/x-mspowerpoint (also represents application/vnd.ms-powerpoint)
Microsoft Visio	application/vnd.visio
Microsoft Word Document	application/msword
Microsoft Works Word Processor Document	application/x-msworks-wp
MS Write	application/x-mswrite
PDF Document	application/pdf
Plain Text	text/plain
Quattro Pro for Windows Document	application/x-quattro-win
Rich Text Format (RTF) Document	application/rtf
StarOffice Calc Document	application/vnd.stardivision.calc
StarOffice Impress Document	application/vnd.stardivision.impress
Sun XML Writer Document	application/vnd.sun.xml.writer
TIF Image	image/tiff
WordPerfect 5.1 Document	application/wordperfect5.1
WordPerfect 6 Document	application/x-wordperfect6
XML	text/xml
XyWrite Document	application/x-xywrite
ZIP Archive	application/zip

Globalization Support

Oracle SES provides localization support for source documents, metadata translation, and user queries. You can specify this information in the configuration of administration objects.

Product Languages

Oracle SES user interface components are translated into the languages listed in [Table 2–2](#). The locale of the Oracle SES host system sets the default language for error messages, as well as the Administration GUI and the Search Application. In the Web services interface, you can set the language for error messages in individual operations.

Table 2–2 Product Languages

Language	Code
Chinese, Simplified	zh_CN
Chinese, Traditional	zh_TW
English	en
French	fr
German	de

Table 2–2 (Cont.) Product Languages

Language	Code
Italian	it
Japanese	ja
Korean	ko
Portuguese, Brazilian	pt_BR
Spanish	es

Crawlable Documents

For Oracle SES to crawl and index source documents, they must be stored in a supported language and character set.

[Table 2–3](#) lists the codes for languages supported by the crawler.

Table 2–3 Crawlable Languages

Language	Code
Arabic	ar
Chinese	zh
Czech	cs
Danish	da
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Greek	el
Hebrew	he
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl
Portuguese	pt
Romanian	ro
Russian	ru
Slovak	sk
Spanish	es
Swedish	sv
Turkish	tr

[Table 2–4](#) lists the codes for character sets supported by the crawler.

Table 2–4 Crawlable Character Sets

Character Set	Code
Standard UTF-8	UTF8
16-Bit UCS Transformation Format	UTF-16
Big 5 Traditional Chinese	Big5
CNS 11643 Traditional Chinese	CNS11643
GB 18030 Simplified Chinese	GB18030
GB2312-80 Simplified Chinese	GB2312
GBK Simplified Chinese	GBK
ISO Latin/Arabic	8859-6
ISO Latin/Cyrillic	8859-5
ISO Latin/Greek	8859-7
ISO Latin/Hebrew	8859-8
ISO Latin-1	8859-1
ISO Latin-2	8859-2
ISO Latin-3	8859-3
ISO Latin-4	8859-4
ISO Latin-5	8859-9
Japanese (Auto-Detect)	JISAutoDetect
Japanese (EUC)	EUC_JP
Japanese (JIS)	JIS
Japanese (Shift-JIS)	SJIS
KSC5601 Korean	KSC5601
Macintosh Arabic	MacArabic
Macintosh Croatian	MacCroatian
Macintosh Cyrillic	MacCyrillic
Macintosh Dingbat	MacDingbat
Macintosh Greek	MacGreek
Macintosh Hebrew	MacHebrew
Macintosh Iceland	MacIceland
Macintosh Latin-2	MacCentralEurope
Macintosh Roman	MacRoman
Macintosh Romania	MacRomania
Macintosh Symbol	MacSymbol
Macintosh Thai	MacThai
Macintosh Turkish	MacTurkish
Macintosh Ukraine	MacUkraine
PC Arabic	Cp864

Table 2–4 (Cont.) Crawlable Character Sets

Character Set	Code
PC Baltic	Cp775
PC Canadian French	Cp863
PC Cyrillic	Cp855
PC Greek	Cp737
PC Hebrew	Cp862
PC Icelandic	Cp861
PC Latin-1	Cp850
PC Latin-2	Cp852
PC Modern Greek	Cp869
PC Nordic	Cp865
PC Original	Cp437
PC Portuguese	Cp860
PC Russian	Cp866
PC Turkish	Cp857
Windows Arabic	Cp1256
Windows Baltic	Cp1257
Windows Cyrillic	Cp1251
Windows Eastern Europe/Latin-2	Cp1250
Windows Greek	Cp1253
Windows Hebrew	Cp1255
Windows Japanese	MS932
Windows Thai	Cp874
Windows Turkish	Cp1254
Windows Vietnamese	Cp1258
Windows Western Europe/Latin-1	Cp1252

Providing Translations of Object Names

The names of some administration objects are displayed to users in the Search interface, such as `source`, `sourceGroup`, and `clusterTree`. You can provide a display name in one or more languages by using the `<search:translations>` element, as shown here:

```
<search:name>
  <search:translations>
    <search:translation>
      <search:translatedValue>
```

Element Descriptions

<search:name>

Name of the administration object.

<search:translations>

Contains one or more <search:translation> elements.

<search:translation>

Contains a <search:translatedValue> element.

Attribute	Value
language	A code identifying the language of the translated value. The codes are not case sensitive. See Table 2-5, "Query Language Codes" .

<search:translatedValue>

Contains a description of the object in the translation language. This value is displayed in the Search Application.

Table 2-5 Query Language Codes

Language	Code
Arabic	ar
Catalan	ca
Chinese, Simplified	zh_CN
Chinese, Traditional	zh_TW
Czech	cs
Danish	da
Dutch	nl
English	en
Finnish	fi
French	fr
German	de
Greek	el
Hebrew	iw
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl
Portuguese	pt
Portuguese, Brazilian	pt_BR
Romanian	ro
Russian	ru
Slovak	sk
Spanish	es
Swedish	sv

Table 2–5 (Cont.) Query Language Codes

Language	Code
Thai	th
Turkish	tr

Encryption

The Administration API provides an encryption system to safeguard sensitive information, such as passwords, contained in the XML description of an object.

When you import an XML document using an operation such as create or update, you can indicate in the XML whether a value is encrypted. In this example, the password is in plain text, which either sets it for the first time or changes it to a new value:

```
<search:password encrypted="false">password</search:password>
```

Oracle SES stores the password in an encrypted form. The next example shows an encrypted password, which was exported in an XML document from Oracle SES:

```
<search:password encrypted="true">
128b6b43091659ffa1ff068666b8eb6445dabd361871b6a5b97941f00ee8c842e76bcc1eb3c0806fd0
f6ee2e3ab371febcf053255ffd4e46888909cdd553914bfabe99eda51861d7
</search:password>
```

When exporting an XML document containing a password, Oracle SES requires you to provide an encryption key. If you use this document as input to an operation (`encrypted="true"`), then you must use the same encryption key as the `export` operation so that Oracle SES can decrypt the password.

XML Description of State Properties

Both universal and creatable objects can have state properties. The `getState`, `getStateList`, and `getAllStates` commands return an XML document describing the current state of one or more objects.

The `<search:state>` element describes the current state of an object.

```
<search:state>
  <search:objectStates>
    <search:objectState>
      <search:objectState>
        <search:objectType>

<!-- For creatable objects -->
        <search:objectKey>
          <search:keyPairs>
            <search:keyPair>
              <search:name>
              <search:value>

<!-- For all objects -->
        <search:stateProperties>
          <search:stateProperty>
            <search:propertyName>
            <search:propertyValues>
              <search:propertyValue>
              <search:propertyValue>
```

Element Descriptions

<search:state>

Contains a <search:objectStates> element.

Attribute	Value
productVersion	Oracle SES product version
xmlns:search	Namespace for the Oracle SES Administration API

<search:objectStates>

Contains one or more <search:objectState> elements.

<search:objectState>

Describes the state properties of a particular object, using these child elements:

```
<search:objectType>
<search:objectKey>
<search:stateProperties>
```

<search:objectType>

Contains an object type with one or more state properties:

```
clustering
clusterTree
identityPlugin
index
indexOptimizer
resultList
schedule
skinBundle
spaceCalculator
task
```

<search:objectKey>

Contains the object key that identifies a specific instance of a creatable object type. It contains a <search:keyPairs> element.

<search:keyPairs>

Contains one or more <search:keyPair> elements.

<search:keyPair>

Contains these child elements:

```
<search:name>
<search:value>
```

<search:name>

Contains a key name for this object type.

<search:value>

Contains the key value for this object.

<search:stateProperties>

Contains one or more <search:stateProperty> elements.

<search:stateProperty>

Contains a <search:propertyName> element.

<search:propertyName>

Contains the name of a property.

<search:propertyValues>

Contains one or more `<search:propertyValue>` elements.

<search:propertyValue>

Contains a `<search:value>` element.

Attribute	Value
key	Provides additional context, such as the name of the data source associated with the property for a schedule that crawls multiple sources.

<search:value>

Contains the current value of the property.

Partitioning for Parallel Query

You can optimize query performance of large document sources by storing the crawler index in partitions distributed across several independent disks. Oracle SES then executes parallel subqueries automatically against the partitions. Both I/O and CPU resources are used in parallel.

These administrative objects support parallel query:

- `partitionConfig`
- `storageArea`

Note: To support parallel query, you must create the partitions immediately after installing Oracle SES. You cannot create them after crawling a document source.

To support parallel query:

1. Open a SQL session as the administrative user:


```
sqlplus eqsys
```
2. Execute two PL/SQL procedures to enable the partitioning feature of the Oracle SES instance:


```
exec eq_adm.use_instance(1)
exec eq_par.enable_partition
```
3. Define one or more `storageArea` objects with a usage of `PARTITION` on each physical storage device available to this instance.
4. Update the `partitionConfig` object to have a rule type of `HASH` and to use the new `storageArea` objects.
5. Create document sources and schedule them for crawling.

See Also: "Parallel Query Indexing" in the *Oracle Secure Enterprise Search Administrator's Guide*.

Parallel Query Example

This example creates two partitions, using the default OES storage area and a newly created OES1 storage area:

1. Using `searchadmin`, export the XML description of the default OES storage area to a file named `oes.xml`:

```
export storageArea --NAME=oes --OUTPUT_FILE=oes.xml
```

2. Export the XML description of the partition configuration to a file named `part.xml`:

```
export partitionConfig --OUTPUT_FILE=part.xml
```

3. Open `oes.xml` in a text editor and edit it as follows:

```
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:storageAreas>
    <search:storageArea>
      <search:name>OES1</search:name>
      <search:description>
        Default storage area extension
      </search:description>
      <search:usage>PARTITION</search:usage>
      <search:locations>
        <search:location>
          <search:path>/ses_storage/</search:path>
          <search:device>default</search:device>
          <search:preAllocatedSpace>20</search:preAllocatedSpace>
          <search:quota>400</search:quota>
        </search:location>
      </search:locations>
    </search:storageArea>
  </search:storageAreas>
</search:config>
```

4. Open `part.xml` in a text editor and edit it as follows:

```
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:partitionConfig>
    <search:partitionRules>
      <search:partitionRule>
        <search:partitionValue>EQ_DEFAULT</search:partitionValue>
        <search:valueType>META</search:valueType>
        <search:ruleType>HASH</search:ruleType>
        <search:storageArea>oes, oes1</search:storageArea>
      </search:partitionRule>
    </search:partitionRules>
  </search:partitionConfig>
</search:config>
```

5. Create the new storage area:

```
create storageArea --NAME=oes1 --INPUT_FILE=oes.xml
```

6. Update the partition configuration:

```
update partitionConfig --INPUT_FILE=part.xml --UPDATE_METHOD=overwrite
```

Disk Space Management: Quotas and Alerts

Oracle SES uses quotas to restrict the amount of disk space that it consumes. These administrative objects set and enforce these quotas:

- `alert`
- `spaceCalculator`
- `storageArea`

A quota is set on the default data storage location as a post- installation step, computed as half the remaining disk space plus its initial size. You can alter this quota and set quotas on other storage areas.

See Also: See "Managing Disk Space Usage" in the *Oracle Secure Enterprise Search Administrator's Guide*

To set a disk space quota:

1. Identify the storage area that you want to create or modify.
2. In the XML description of the object, set the `<search:quota>` element to the maximum desired size in megabytes.
3. Use the `create`, `createAll`, `update`, or `updateAll` operation to create or modify the object.
4. Configure the space calculator to run at the desired intervals.

The space calculator is set initially to run daily and is activated automatically. It issues a warning if a quota is 80% filled, or an alert if a quota is exceeded.

You can see alerts on the Home - General page of the Administration GUI, or by issuing an `exportAll alert` operation. You must clear an alert before normal operations can resume.

To clear an alert:

1. Remove the cause of the alert, such as by increasing the quota or removing files from the storage area.
2. Start the space calculator to check the storage area size against the quota again.

The space calculator changes the alerts to `RESOLVED` if the storage area is within the quota. Otherwise, the alert remains `OPEN`.

3. Start the `resumeAllSpaceConsumingTasks` task to resume normal operation.

Tip: You can delete crawler logs and cache files. If this does not free enough space, you can create a storage area on another disk for the crawler to use. The crawler stores data in Oracle data files (*.dbf), which you should not delete at the operating system level.

To remove a storage area quota:

1. In the XML description of the object, set the `<search:quota>` element to `-1`.
2. Use the `update` or `updateAll` operation to modify the object.

Space Management Examples

This example changes the default quota on the data storage location:

1. Export the data storage object to a file named `datastore.xml`:

```
export storageArea --NAME="Data storage location" --OUTPUT_FILE=datastore.xml
```

2. Open datastore.xml in an XML or text editor and change the value of the `<search:quota>` element as desired. This example sets it to 64 GB (64*1024 MB).

```
<search:quota>65536</search:quota>
```

3. Save and close datastore.xml.
4. Update the changes for this storage area:

```
update storageArea --NAME="Data storage location" --INPUT_FILE=datastore.xml
--UPDATE_METHOD=overwrite
```

5. Check the current disk usage:

```
start spaceCalculator
```

The next example clears an alert signalled by the space calculator.

1. Fix the cause of the alert. In this example, a new Web source was defined with an unlimited crawling depth. The solution was to limit the crawling depth of the Web source and increase the quota on the data storage location to accommodate the additional crawled documents.

2. Start the space calculator:

```
start spaceCalculator
```

3. Check the status of the alert on the Oracle Administration GUI or by exporting the alert:

```
export alert --NAME=alert_1
```

This XML description shows that the problem is resolved:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:alerts>
    <search:alert>
      <search:name>alert_1</search:name>
      <search:time>Thu, 14 Jan 2010 15:38:21 GMT</search:time>
      <search:type>SPACE</search:type>
      <search:status>RESOLVED</search:status>
      <search:causes>
        <search:cause location="/home/oracle/dbs/sesmain/" quota="4"
size="293" storageArea="Data storage location" time="14 Jan 2010 09:38:21"/>
      </search:causes>
    </search:alert>
  </search:alerts>
</search:config>
```

4. Resume normal operations:

```
start task --NAME=resumeAllSpaceConsumingTasks
```

Search Interface Customization: Skin Bundles

You can alter the look and feel of the Search application by creating a custom "skin" -- or user interface -- with different graphics, fonts, and colors. The files composing a custom skin are called, collectively, a **skin bundle**.

Support Bundles

All of the files associated with the Search application user interface for a particular release are supplied in a **support bundle**. These files include FreeMarker templates, images, style sheets, and JavaScript libraries.

The templates that you modify or replace are included in your skin bundle. When Oracle SES does not find a template file in the skin bundle that is needed to display a page in the Search application, then it uses the template file in the support bundle.

Both support bundles and skin bundles are associated with a particular release. This association enables you to migrate skin bundles to future releases of Oracle SES, even though the default user interface might change. When rendering the Search application pages, Oracle SES can still combine files from the skin bundle with files in the support bundle for the same release.

The current support bundle is located in this directory:

```
ORACLE_HOME/search/data/queryapp/support/11_1_2_0_0
```

FreeMarker Templates

FreeMarker is an open-source tool that generates text from templates. The templates replace HTML files for generating a page in a browser. Oracle SES uses FreeMarker to isolate the look-and-feel of the Search Application from the search software.

The FreeMarker templates are located in the templates directory of the support bundle and have an ftl extension to the file name, such as templates/results.ftl. Before editing the template files, you should become familiar with FreeMarker.

See Also: FreeMarker Web site at
<http://www.freemarker.org/>.

The templates contain HTML and two other types of tags:

- **FreeMarker tags:** These tags are predefined in FreeMarker and begin with <#. For example, this tag appears at the beginning of most templates:

```
<#import "/lib/oracle.com/seslib.ftl" as ses>
```

The FreeMarker Manual describes these tags, which invoke predefined directives, at http://freemarker.org/docs/ref_directives.html.

- **Oracle SES tags:** These tags are specific to Oracle SES and begin with <@. For example, this tag references a graphic file named logo.gif in the skin bundle:

```
<@ses.skin_asset 'images/logo.gif' />
```

Oracle SES tags invoke macros (also called user-defined directives) defined in seslib.ftl, so any template that uses them must import that file. The *Oracle Secure Enterprise Search Administrator's Guide* describes these macros.

Asset Files

Cascading style sheets, graphics, and JavaScript files are **assets**. You can revise an asset file from the support bundle like a template file, or you can create your own custom asset files.

When using custom asset files, you must include references to them using macros within standard HTML. For example, you might create a style sheet named mystyles.css with redefined tags from the support bundle, then include it in your skin

bundle templates with a tag like the following. Note the use of the `<@ses.skin_asset>` macro, which identifies the location of `mystyles.css` in the skin bundle.

```
<link rel="stylesheet" type="text/css" href="<@ses.skin_asset
filename='css/mystyles.css' />">
```

Similarly, the next tag references a graphics file named `mylogo.gif`:

```

```

Alternatively, you might copy `search.css` and `oraclelogo_medium.gif` into your skin bundle and modify their contents. Then you would modify references to these files to use the `<@ses.skin_asset>` macro, which points to the version of the asset in your skin bundle instead of the file in the support bundle.

Tip: To trace the styles formatting a particular element on the page, use the development tools of your browser, such as the Firebug extension to Mozilla Firefox, the Inspect Element tool in Google Chrome, or the Developer Toolbar extension to Microsoft Internet Explorer.

JavaScript Libraries

The Oracle SES 11.1.2.2.0 support bundle contains two JavaScript libraries:

- **Yahoo! User Interface (YUI) Library:** A set of utilities and controls for building interactive Web applications.
- **Bubbling Library extension to YUI:** A set of plug-ins and widgets.

See Also:

- YUI Library section of the Yahoo! Developer Network site at <http://developer.yahoo.com/yui/>.
- Bubbling Library Web site at <http://bubbling-library.com/>.

Template Library

The support library contains a file named `seslib.ftl` that references all of the resources available to the templates: JavaScript files, style sheets, macros, and so forth. The Freemaker templates import `seslib.ftl` using this tag at the top of each file:

```
<#import "/lib/oracle.com/seslib.ftl" as ses>
```

The tag makes these resources available for use in the template. You can delete the tag if you do not need these resources to generate a particular page, but do not modify the file.

Configuration Settings

Configuration settings for the Search application are stored in `<variable>` elements in this file:

```
ORACLE_HOME/search/tools/weblogic/deploy/plans/QueryPlan.xml
```

The *Oracle Secure Enterprise Search Administrator's Guide* describes these settings. After making any changes, issue the following shell command from Linux and UNIX-based operating systems:

```
sh $ORACLE_HOME/search/tools/weblogic/deploy/deployer.sh -serverURL
```

```
t3://host:port/ -user weblogic -password password -name search_query -plan
$ORACLE_HOME/search/tools/weblogic/deploy/plans/QueryPlan.xml -process redeploy
```

Or run this batch file from Windows:

```
%ORACLE_HOME%\search\tools\weblogic\deploy\deployer.bat -serverURL t3://host:port/
-user weblogic -password password -name search_query -plan %ORACLE_
HOME%\search\tools\weblogic\deploy\plans\QueryPlan.xml -process redeploy
```

Where:

host:port is the host name and WebLogic service port, such as `example:7777`. This is the same port that you use to open the Administration GUI.

password is the password for `eqsys`.

Assembling the Skin Bundle Files

To assemble the skin bundle files:

1. Decide on the changes to make to the Search application, such as replacing the logo or the icons, changing the default font or background color, or adding an RSS feed.
2. Create the following directory structure for storing the files composing the skin bundle:

```
/skinBundle_name
  /templates
  /assets
    /images
    /css
    /js
```

3. Identify the template files that render the changed pages.

For descriptions of the template files, see the *Oracle Secure Enterprise Search Administrator's Guide*.

4. Copy the `ftl` files from the support bundle for the current release of Oracle SES into the `templates` directory. Do not change the names of these files.
5. Modify the templates as desired, using a text editor. Templates can include HTML tags, FreeMarker tags, and Oracle SES tags. You can change text and various settings, and reference custom graphics, style sheets, and JavaScript. See "[FreeMarker Templates](#)" on page 2-14.
6. Create the graphic files, cascading style sheets, and JavaScript files as desired. Copy the graphics files into the `images` directory, the cascading style sheets into the `css` directory, and the JavaScript files into the `js` directory.
7. Create an XML document that describes the skin bundle. See [skinBundle](#) on page 2-66.

Creating a skinBundle Object

To create a `skinBundle` object using the command-line API:

1. Assemble the files composing the skin bundle, as previously described.
2. Create a text file that lists all of the files in the skin bundle. See the Notes for [create skinBundle](#) on page 3-18.

3. Issue a `create` command to create the `skinBundle` object.

To create a `skinBundle` object using the Web service API:

1. Assemble the files composing the skin bundle, as previously described.
2. Compose the SOAP message for a `create` operation, as described in [Chapter 4, "Web Service Operations."](#) Include an `<attachments>` element for each file in the skin bundle.
3. Submit the request to the Web service to create the `skinBundle` object.

To create a `skinBundle` object using the Java client, see the *Oracle Secure Enterprise Search Java API Reference*.

Using a Skin Bundle to Render the Search Application User Interface

To use a skin bundle when rendering the Search interface:

1. Issue an `activate` operation for the `skinBundle`. When you activate a default skin bundle, it can be used immediately to render the Search Application interface.
2. To use a skin bundle that is not the default, add a `skin=skin_name` attribute to the URL for the Search Application interface:

```
http://host:port/search/query/search?skin=skin_name
```

If the modified pages fail to open in a browser or appear with errors, read the middle-tier log file at

```
ORACLE_HOME/search/base_
domain/servers/AdminServer/logs/AdminServer.out.
```

After updating the skin bundle, restart the middle tier:

```
ORACLE_HOME/bin/searchctl restart
```

Skin Bundle Example

This example makes a few changes to the default results page, which is shown in [Figure 2-1](#).

Changes to the Example Results Page

[Table 2-6](#) identifies the changes that this example makes to the default results page. You can see these differences by comparing [Figure 2-1](#) and [Figure 2-2](#). The title in the browser title bar is not shown.

Changes to `results.ftl` do not affect any other pages of the Search application, which continue to use the default skin. However, the example makes changes to `inc_logo_querybox.ftl` and `inc_footer.ftl`, which affect all of the pages that include those templates.

Table 2-6 Differences Between the Default Skin and the Example Skin

Default Skin	Example Skin	Template Rendering the Element
Oracle logo	Example Inc. logo	<code>inc_logo_querybox.ftl</code>
Search button	Search icon	<code>inc_logo_querybox.ftl</code>
Sidebar on left	Sidebar on right	<code>results.ftl</code>

Table 2–6 (Cont.) Differences Between the Default Skin and the Example Skin

Default Skin	Example Skin	Template Rendering the Element
Title of Oracle Secure Enterprise Search	Title of Example Inc.	results.ftl
No RSS feed	RSS feed icon on the Results bar	results.ftl
No corporate identifier	Example, Inc. above the copyright	inc_footer.ftl

Figure 2–1 Default Results Page

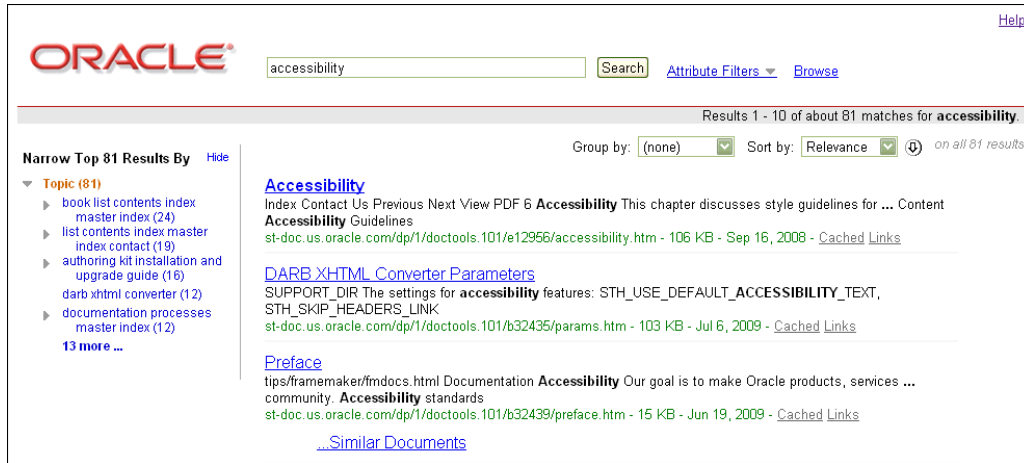
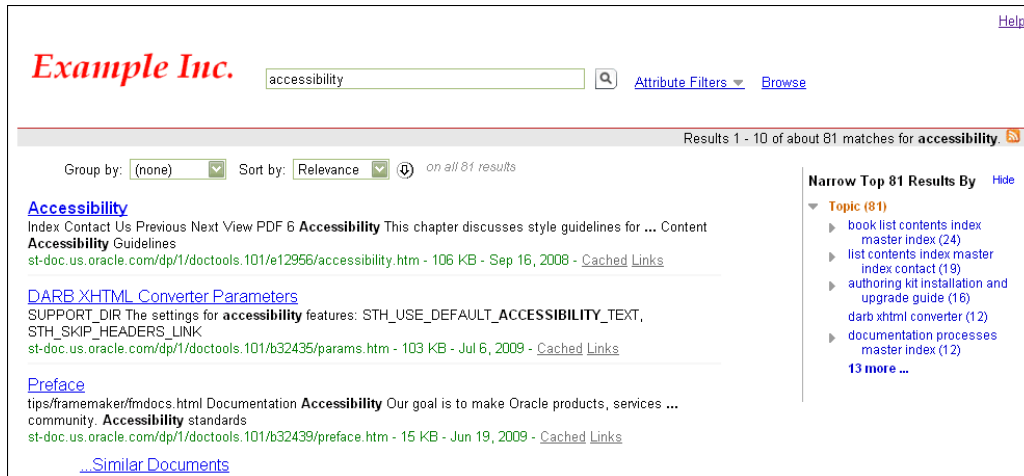


Figure 2–2 Example Results Page



Changes to the Example Footer

The only change to the footer is the addition of *Example Inc.*, as shown in Figure 2–3. The following pages use the same footer template, so all of them are affected by this change:

- Initial splash screen: query.ftl
- Results page: results.ftl
- No results page: noresults.ftl

- Error page: error.ftl

Figure 2–3 Example Footer



Creating the Example Directory Structure

To make the changes to the skin shown in the previous section, the skin bundle must contain these files:

- **results.ftl**: The template that renders the search results.
- **inc_logo_querybox.ftl**: A template included by results.ftl to generate the logo and the query box.
- **inc_footer.ftl**: A template included by results.ftl (and other templates) to generate the footer.
- **example.gif**: A graphic file with the logo for a fictitious company named Example Inc.
- **search.jpg**: a graphic file with the search icon.
- **rss.jpg**: A graphic file with the standard RSS icon.

To create the example skin bundle directory structure:

1. On the Oracle SES host, create these directories:

```
/example/templates
/example/assets/images
```

2. Copy the ftl files to the templates directory from

```
ORACLE_HOME/data/queryapp/support/11_1_2_0_0/templates
```

3. Copy the graphics file (created or acquired elsewhere) into the images directory.

The resulting directories have this structure:

```
/example
  /templates
    /inc_footer.ftl
    /inc_logo_querybox.ftl
    /results.ftl
  /assets
    /images
      /example.gif
      /rss.jpg
      /search.jpg
```

Customizing results.ftl

The results page contains numerous elements. Some elements appear by default, while you must define others, such as source groups and suggested links, for a specific installation. The results.ftl template uses the FreeMarker `<#include>` tag to include the following template files, which define distinct areas of the results page:

- inc_header.ftl

- inc_logo_querybox.ftl
- inc_footer.ftl

This example uses the default inc_header.ftl, but alters the other templates. [Figure 2–6](#) identifies the altered elements that are generated directly by results.ftl.

To customize results.ftl:

1. Open example/templates/results.ftl in a text editor.
2. To move the sidebar to the right, *change*:

```
<#assign sidebarPageAlign = "left">

to

<#assign sidebarPageAlign = "right">
```

3. To replace the page title, *change*:

```
<title>${msg("ORACLE_ENTERPRISE_SEARCH")}
  <#if req.displayQuery??>
    - ${req.displayQuery}
  </#if>
</title>

to

<title>Example Inc.</title>
```

4. For the RSS feed, *add* the following immediately after <@ses.hit_stats/>:

```
<#assign feed_img_src<@ses.skin_asset 'images/rss.jpg' /></#assign>
<@ses.feed_icon title="Results Feed" img_src="${feed_img_src}">
  <@ses.feed_href/>
</@ses.feed_icon>
```

5. Save and close the file.

Customizing inc_logo_querybox.ftl

The inc_logo_querybox.ftl template renders a section of the results page immediately following the header. This section includes these elements in the default user interface:

- Oracle logo
- Query box
- Search button
- Attribute filters, both the link and the form
- Browse link
- Optional source group tab links above the query box, such as E-mail, Calendar, and Sales.

To customize inc_logo_querybox.ftl:

1. Open example/templates/inc_logo_querybox.ftl in a text editor.
2. To replace the Oracle logo with the Example logo, *change*:

```
<@ses.oracle_logo size="small" href="${logoHref}"/>

to
```

```

```

3. To replace the Search button with an icon, *change*:

```
<input type="submit" name="btnSearch" value="{msg("SEARCH")}">
```

to

```
<input type="image" src="@ses.skin_asset filename="images/search.jpg" />"
name="{msg("SEARCH")}" alt="{msg("SEARCH")}"
style="vertical-align: bottom;">
```

4. Save and close the file.

Customizing inc_footer.ftl

The inc_footer.ftl template renders the links, such as Help, and the copyright information at the bottom of the page.

To customize inc_footer.ftl:

1. Open example/templates/inc_footer.ftl in a text editor.
2. For the company name, *add* the following immediately before `<!-- Bottom Line -->`:

```
<div style="padding-top:10px;font-size:16px;font-weight:bold;
font-style:italic;color:red;font-family:'Book Antigua',Palatino,serif;
text-align:center">
    Example Inc.
</div>
```

3. Save and close the file.

Creating the Example Skin Bundle File List

Create a text file that identifies all of the files in the skin bundle. In this example, the file list is named /scratch/skins/example.lst. Substitute the actual path you are using for /scratch/skins.

```
assets/images/example.gif::scratch/skins/example/assets/images/example.gif
assets/images/search.jpg::scratch/skins/example/assets/images/search.jpg
assets/images/rss.jpg::scratch/skins/example/assets/images/rss.jpg
templates/inc_footer.ftl::scratch/skins/example/templates/inc_footer.ftl
templates/inc_logo_querybox.ftl::scratch/skins/example/templates/inc_logo_
querybox.ftl
templates/results.ftl::scratch/skins/example/templates/results.ftl
```

Creating an XML Description of the Example Skin Bundle

Create an XML file that describes the Example skin bundle. In this example, the XML file is named /scratch/skins/example.xml.

```
<?xml version="1.0" encoding="UTF-8" ?>

<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:skinBundles>
    <search:skinBundle>
      <search:name>example</search:name>
      <search:isDefault>>false</search:isDefault>
      <search:linkedVersion>11.1.2.2.0</search:linkedVersion>
      <search:files>
```

```
<search:file path="templates/inc_footer.ftl"/>
<search:file path="templates/inc_logo_querybox.ftl"/>
<search:file path="templates/results.ftl"/>
<search:file path="assets/images/example.gif"/>
<search:file path="assets/images/search.jpg"/>
<search:file path="assets/images/rss.jpg"/>
</search:files>
</search:skinBundle>
</search:skinBundles>
</search:config>
```

Creating the Example skinBundle Object

To create the Example skin bundle:

1. At the host command prompt, navigate to the /scratch/skins directory.
2. Open `searchadmin` in session mode, as described in ["Opening an Interactive Session"](#) on page 1-2.
3. To create the skin bundle, issue this command:

```
create skinBundle --NAME=example --INPUT_FILE=example.xml --ATTACHMENT_
LIST=example.lst
```

4. To activate the skin bundle, issue this command:

```
activate skinBundle --NAME=example
```

Using the Example Skin Bundle to Render the Search Application

Because the example skin bundle is not defined as the default, you must include the skin attribute in the URL to view the Search application.

To use the Example skin bundle:

1. In a browser, enter a URL like the following, substituting the appropriate host and port:

```
http://host:port/search/query/search?skin=example
```

The footer displays Example Inc., while the rest of the page uses the default skin.

2. Enter a search string. The results page has the changes shown in [Figure 2-2, "Example Results Page"](#).

alert

An alert is a warning that a storage area has reached its maximum size. This alert appears on the Oracle SES Administration GUI home page. All Oracle SES background processes that consume disk space are stopped until you clear the alert.

See Also: ["Disk Space Management: Quotas and Alerts"](#) on page 2-12

Object Type

Creatable by the space management system

Object Key

name

Object Key Command Syntax

```
--NAME=object_name
```

```
-n object_name
```

State Properties

None

Supported Operations

```
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll
```

Administration GUI Page

Home - General (display only)

XML Description

The `<search:alerts>` element describes alerts:

```
<search:alerts>
  <search:alert>
    <search:name>
    <search:time>
    <search:type>
    <search:status>
    <search:causes>
      <search:cause>
```

Element Descriptions

`<search:alerts>`

Contains one or more `<search:alert>` elements.

<search:alert>

Describes an alert. It contains these elements:

```
<search:name>
<search:time>
<search:type>
<search:status>
<search:causes>
```

<search:name>

Contains the name of the alert.

<search:time>

Contains the date and time the alert was raised.

<search:type>

Contains the type of alert:

- **SPACE:** The storage area filled its space quota. Tasks are stopped until this alert is resolved.
- **SPACE_WARNING:** The storage area filled 80% or more of its space quota. No tasks are stopped; this is just a notice.

<search:status>

Contains the current status of the alert. This is the only setting that you can update.

- **OPEN:** The alert is open and Oracle SES processes are stopped. The system typically sets this status when creating the alert.
- **RESOLVED:** The condition that caused the alert has been fixed and processing can resume. You can set this status with an `update` or `updateAll` operation.

<search:causes>

Contains one or more `<search:cause>` elements.

<search:cause>

Describes the storage area that caused the alert.

Attribute	Value
time	Date and time the alert was raised.
location	Directory location of the storage area.
storageArea	Name of the storage area.
size	Current size of the storage area.
quota	Maximum size of the storage area.

Examples

This XML document shows an alert on the OES and OES_ASSM2 storage areas:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:alerts>
    <search:alert>
      <search:name>alert_1</search:name>
      <search:time>Wed, 09 Dec 2009 19:46:32 GMT</search:time>
      <search:type>SPACE</search:type>
```

```
<search:status>OPEN</search:status>
<search:causes>
  <search:cause location="/home/oracle/dbs/ses/" quota="2" size="140"
storageArea="OES" time="09 Dec 2009 13:46:32"/>
  <search:cause location="/home/oracle/dbs/ses/" quota="2" size="111"
storageArea="OES_ASSM2" time="09 Dec 2009 13:46:32"/>
</search:causes>
</search:alert>
</search:alerts>
</search:config>
```

altWord

Oracle SES uses alternate words to provide suggestions to users or to expand the search results. Alternate words are useful for correcting common typing errors and for including synonyms in a search. You can create up to four alternates for the same word.

Object Type

Creatable

Object Key

keyword altKeyword

Object Key Command Syntax

```
--KEYWORD=keyword --ALT_KEYWORD=altKeyword
```

State Properties

None

Supported Operations

```
create  
createAll  
delete  
deleteAll  
deleteList  
export  
exportAll  
exportList  
getAllObjectKeys  
update  
updateAll
```

Administration GUI Page

Search - Alternate Words

XML Description

The `<search:altWords>` element describes alternate word pairs:

```
<search:altWords>  
  <search:altWord>  
    <search:keyword>  
    <search:altKeyword>  
  <search:autoExpand>
```

Element Descriptions

`<search:altWords>`

Contains one or more `<search:altWord>` elements.

`<search:altWord>`

Contains one of each of these elements:


```
<search:keyword>
<search:altKeyword>
<search:autoExpand>
```

<search:keyword>

Contains a search word or phrase. Keywords are not case sensitive. Required.

<search:altKeyword>

Contains a word or phrase that is suggested when users enter the keyword. Alternate words are displayed exactly as they appear here. Required.

<search:autoExpand>

Controls the display of alternative words in the search results: Set to `true` to include the alternative words automatically in the search, or set to `false` to display alternative word matches in a "do you mean..." message. The default is `false`.

Example

This XML document defines alternate words for OSES, text, and RAC:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:altWords>
    <search:altWord>
      <search:keyword>oses</search:keyword>
      <search:altKeyword>Oracle Secure Enterprise Search</search:altKeyword>
      <search:autoExpand>true</search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>rac</search:keyword>
      <search:altKeyword>Real Application Clusters</search:altKeyword>
      <search:autoExpand>>false</search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>text</search:keyword>
      <search:altKeyword>Oracle Text</search:altKeyword>
      <search:autoExpand>>false</search:autoExpand>
    </search:altWord>
  </search:altWords>
</search:config>
```

clustering

Query-time clustering dynamically organizes search results into groups to provide end users with different views of the top results. Clustered documents within one group, called a cluster node, share the same common topics or property values. A cluster node for a large document set can be categorized into child cluster nodes, creating a hierarchy. Users can navigate directly to a specific cluster node. Effective real-time clustering balances clustering quality and clustering time.

Object Type

Universal

State Properties

Property	Value
status	ACTIVE INACTIVE

Supported Operations

activate
deactivate
export
getState
update

Administration GUI Page

Global Settings - Query-Time Clustering Configuration

XML Description

The `<search:clustering>` element describes clustering:

```
<search:clustering>
  <search:maxTreeDepth>
  <search:maxChildrenPerNode>
  <search:minDocsPerNode>
  <search:minOccurrenceWords>
  <search:maxExtractWords>
  <search:minOccurrencePhrases>
  <search:maxExtractPhrases>
  <search:maxPhraseLength>
```

Element Descriptions

`<search:clustering>`

Contains all of the elements for clustering parameters, which are described in the following paragraphs.

`<search:maxTreeDepth>`

Maximum number of levels in a cluster node hierarchy. (Optional)

A cluster node with a large document set can be categorized into child cluster nodes. A cluster hierarchy gives end users a quick overview of the results. They can navigate

directly to a specific cluster node or refine their query by combining the original query and cluster results.

<search:maxChildrenPerNode>

Maximum number of cluster nodes on each level.

<search:minDocsPerNode>

Minimum number of documents in a cluster node.

<search:minOccurrenceWords>

Minimum occurrences of a word to be extracted for topic clustering.

<search:maxExtractWords>

Maximum number of words to be extracted for topic clustering.

<search:minOccurrencePhrases>

Minimum occurrences of a phrase to be extracted for topic clustering.

<search:maxExtractPhrases>

Maximum number of phrases to be extracted for topic clustering.

<search:maxPhraseLength>

Maximum word length of phrases to be extracted for topic clustering.

Example

This XML document configures clustering:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:clustering>
    <search:maxTreeDepth>7</search:maxTreeDepth>
    <search:maxChildrenPerNode>125</search:maxChildrenPerNode>
    <search:minDocsPerNode>1</search:minDocsPerNode>
    <search:minOccurrenceWords>9</search:minOccurrenceWords>
    <search:maxExtractWords>18</search:maxExtractWords>
    <search:minOccurrencePhrases>4</search:minOccurrencePhrases>
    <search:maxExtractPhrases>21</search:maxExtractPhrases>
    <search:maxPhraseLength>7</search:maxPhraseLength>
  </search:clustering>
</search:config>
```

clusterTree

Clusters provide users with a tree structure to navigate the top n results by organizing search results into groups. Documents in the same group share the same common topics or property values. Effective real-time clustering balances clustering quality and clustering time.

Clustering does not change the order of the documents. When users select a cluster, the result view is limited to the documents in that cluster. All operations, such as sorting or next page, are limited to the cluster.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

Property	Value
status	ACTIVE
	INACTIVE

Supported Operations

activate
create
createAll
deactivate
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
getAllStates
getState
getStateList
update
updateAll

Administration GUI Page

Global Settings - Clustering Configuration - Create or Edit Metadata Clustering Tree
Global Settings - Clustering Configuration - Create or Edit Topic Clustering Tree

XML Description

The `<search:clusterTrees>` element describes topic cluster trees, metadata cluster trees, or both:

```
<search:clusterTrees>
  <search:topicClusterTree>
    <search:name>
    <search:translations>
    <search:clusteringAttrs>
      <search:clusteringAttr>
        <search:name>

  <search:metadataClusterTree>
    <search:name>
    <search:translations>
    <search:clusteringAttrs>
      <search:clusteringAttr>
        <search:name>
        <search:type>
    <search:tokenized>
    <search:tokenDelimiter>
    <search:hierarchical>
    <search:hierarchyDelimiter>
```

Element Descriptions

`<search:clusterTrees>`

Contains one or more `<search:topicClusterTree>` elements, `<search:metadataClusterTree>` elements, or both.

`<search:topicClusterTree>`

Describes a topic cluster tree. It contains these elements:

```
<search:name>
<search:translations>
<search:clusteringAttrs>
```

`<search:metadataClusterTree>`

Describes a metadata cluster tree. It contains these elements:

```
<search:name>
<search:translations>
<search:clusteringAttr>
<search:tokenized>
<search:tokenDelimiter>
<search:hierarchical>
<search:hierarchyDelimiter>
```

`<search:name>`

Contains the unique name of the cluster tree. Required.

`<search:translations>`

Contains one or more translations of the object name. See ["Providing Translations of Object Names"](#) on page 2-6.

`<search:clusteringAttrs>`

Contains one or more `<search:clusteringAttr>` elements.

<search:clusteringAttr>

Contains a <search:name> element and, for metadata trees, a <search:type> element.

These attributes can be default search attributes, custom search attributes, or Oracle SES internal attributes. Topic tree attributes are String only. For metadata trees, you must specify the data type.

<search:name>

Contains the search attribute used to generate the tree.

<search:type>

Contains the data type of the attribute values. Set to `STRING`, `NUMBER`, or `DATE`.

<search:tokenized>

Controls tokenizing of a String attribute value in a metadataClusterTree. Set to `true` to separate the string into several values where indicated by a delimiter, or set to `false` to handle the string as a single value.

<search:tokenDelimiter>

Identifies the delimiter used to separate tokens in a String attribute value. Set to a character, such as a comma (,) or a hash mark (#). The default delimiter is whitespace (). The token delimiter must be different from the hierarchy delimiter when both are used.

<search:hierarchical>

Controls whether a metadata cluster tree for String attributes has a hierarchical structure. Set to `true` to generate the tree based on a hierarchy implicit in the attribute values, or set to `false` to generate the tree without a hierarchy.

<search:hierarchyDelimiter>

Identifies the delimiter used to separate the categories in a hierarchy for a metadata cluster tree. Set to a character, such as a slash (/). The default delimiter is whitespace (). The hierarchy delimiter must be different from the token delimiter when both are used. Tokens are parsed before the hierarchy.

The following example shows a comma-delimited tokens, and both tokens have a three-level, slash-delimited hierarchy:

```
java/j2ee/jdbc, oracle/search/connector
```

Example

This XML document defines both a topic cluster tree and a metadata cluster tree:

```
<?xml version="1.0" encoding="UTF-8" ?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:clusterTrees>
    <search:topicClusterTree>
      <search:name>Topic Tree</search:name>
      <search:translations>
        <search:translation language="es">
          <search:translatedValue>Árbol del Asunto
        </search:translatedValue>
        </search:translation>
      </search:translations>
      <search:clusteringAttrs>
        <search:clusteringAttr>
          <search:name>eqtopphrases</search:name>
```

```
</search:clusteringAttr>
<search:clusteringAttr>
  <search:name>eqsnippet</search:name>
</search:clusteringAttr>
</search:clusteringAttrs>
</search:topicClusterTree>
<search:metadataClusterTree>
  <search:name>Metadata Tree</search:name>
  <search:translations>
    <search:translation language="es">
      <search:translatedValue>Árbol de los Meta Datos
    </search:translatedValue>
    </search:translation>
  </search:translations>
<search:clusteringAttr>
  <search:name>Infosource</search:name>
  <search:type>STRING</search:type>
</search:clusteringAttr>
<search:tokenized>>true</search:tokenized>
<search:tokenDelimiter>,</search:tokenDelimiter>
</search:metadataClusterTree>
</search:clusterTrees>
</search:config>
```

crawlerSettings

This object configures the global crawler settings that are used by default for new data sources. You can also configure the crawler settings for individual sources, as described in [source](#) on page 2-68.

The Oracle SES crawler is a Java process activated by a schedule. When activated, the crawler spawns a configurable number of processor threads that fetch information from various sources and index the documents. This index is used for searching sources.

Object Type

Universal

State Properties

None

Supported Operations

export
update

Administration GUI Page

Global Settings - Crawler Configuration

XML Description

The `<search:crawlerSettings>` element describes the crawler settings:

```
<search:crawlerSettings>  
  <search:numThreads>  
  <search:numProcessors>  
  <search:crawlDepth>  
    <search:limit>  
  <search:languageDetection>  
  <search:defaultLanguage>  
  <search:crawlTimeout>  
  <search:maxDocumentSize>  
  <search:defaultCharset>  
  <search:cacheDirectory>  
  <search:preserveDocumentCache>  
  <search:servicePipeline>  
    <search:pipelineName>  
  <search:verboseLogging>  
  <search:logDirectory>  
  <search:logLanguage>
```

Element Descriptions

`<search:crawlerSettings>`

Contains all of the elements for configuring the crawler.

`<search:numThreads>`

Contains the number of processes the crawler starts to crawl sources.

<search:numProcessors>

Contains the number of CPUs (or cores in a multi-core processor) on the computer where the crawler runs. This setting determines the optimal number of processes used for document conversion. A document conversion process converts formatted documents into HTML documents for indexing.

<search:crawlDepth>

Controls whether crawling is limited to the number of nested links set by `<search:limit>`.

Attribute	Value
haslimit	Set to <code>true</code> to restrict crawling to the depth limit, or set to <code>false</code> otherwise. Required.

<search:limit>

Contains the number of nested links the crawler follows. Crawling depth starts at 0, so that the crawler only fetches the starting URL. With a crawling depth of 1, the crawler also fetches any document that it linked from the starting URL, and so forth.

<search:languageDetection>

Controls whether the crawler attempts to detect the language of documents that do not specify the language in their metadata.

Language detection involves these steps:

1. The crawler determines the language code by checking the HTTP header `content-language` or the `LANGUAGE` column of a table source.
2. If the crawler cannot determine the language, then the language recognizer attempts to determine a language. The language recognizer operates on the Latin-1 alphabet and any language with a deterministic Unicode range of characters, such as Chinese, Japanese, and Korean.
3. If the language recognizer cannot identify the language, then the default language is used.

Attribute	Value
enabled	Set to <code>true</code> to attempt to detect a language, or set to <code>false</code> to use the default language. Required.

<search:defaultLanguage>

Contains the code for the default language. The default language is used when language detection is disabled or when the crawler and language detector cannot determine the document language. See [Table 2-3, "Crawlable Languages"](#).

<search:crawlTimeout>

Contains the number of seconds allowed for the crawler to access a document.

<search:maxDocumentSize>

Contains the maximum document size in megabytes. Larger documents are not crawled.

<search:defaultCharset>

Contains the default character set. The crawler uses this character set for indexing documents when the character set cannot be determined. See [Table 2-4, "Crawlable Character Sets"](#).

<search:cacheDirectory>

Contains the absolute path to the cache directory. This directory is used only for older data sources that have not been migrated to use the secure cache in Oracle Database. The default path is `oradata/oracle_sid/cache`.

<search:preserveDocumentCache>

Controls whether the cache is saved after indexing.

Attribute	Value
enabled	Set to <code>true</code> to preserve the cache, or set to <code>false</code> to discard it. Required.

<search:servicePipeline>

Controls use of a document service pipeline. A document service pipeline is used for search result clustering. If your installation does not use result clustering for any source, then disable the pipeline.

Attribute	Value
enabled	Set to <code>true</code> to enable the pipeline, or set to <code>false</code> to disable it. Required.

<search:pipelineName>

Contains the name of the document service pipeline used when the pipeline is enabled.

<search:verboseLogging>

Controls the level of detail in logging messages.

Logging everything can create very large log files when crawling a large number of documents. However, in certain situations, it can be beneficial to configure the crawler to record detailed activity.

The crawler maintains the last seven versions of its log file. The format of the log file name is `ids.MMDDhhmm.log`, where *i* is a system-generated ID, *ds* is the source ID, *MM* is the month, *DD* is the date, *hh* is the launching hour in 24-hour format, and *mm* is the minutes. For example, if a schedule for source 23 is launched at 10 pm, July 8th, then the log file name is `i3ds23.07082200.log`. Each successive schedule launching has a unique log file name. When the total number of log files for a source reaches seven, the oldest log file is deleted.

Attribute	Value
enabled	Set to <code>true</code> to record all information, or set to <code>false</code> to record only summary information. Required.

<search:logDirectory>

Contains the directory that stores the crawler log files.

<search:logLanguage>

Contains the language code for messages written to the log file. See [Table 2-3, "Crawlable Languages"](#).

Example

This XML document configures the crawler:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:crawlerSettings>
    <search:numThreads>5</search:numThreads>
    <search:numProcessors>3</search:numProcessors>
    <search:crawlDepth haslimit="true">
      <search:limit>2</search:limit>
    </search:crawlDepth>
    <search:languageDetection enabled="true"/>
    <search:defaultLanguage>en</search:defaultLanguage>
    <search:crawlTimeout>30</search:crawlTimeout>
    <search:maxDocumentSize>10</search:maxDocumentSize>
    <search:defaultCharSet>8859_1</search:defaultCharSet>
    <search:cacheDirectory>oracle/dbs/ses111/cache/</search:cacheDirectory>
    <search:preserveDocumentCache enabled="true"/>
    <search:servicePipeline enabled="true">
      <search:pipelineName>Default pipeline</search:pipelineName>
    </search:servicePipeline>
    <search:verboseLogging enabled="true"/>
    <search:logDirectory>oracle/dbs/ses111/log/</search:logDirectory>
    <search:logLanguage>en-US</search:logLanguage>
  </search:crawlerSettings>
</search:config>
```

identityPlugin

An identity plug-in provides an interface between Oracle Secure Enterprise Search and an identity management system to validate and authenticate users. An identity plug-in is required for secure searches. Secure searches return only the results that the user is allowed to view based on access privileges.

Only one identity plug-in can be active. It is responsible for all authentication and validation activity in Oracle SES. See "[activate identityPlugin](#)" on page 3-6.

Object Type

Creatable

Object Key

jarFilePath managerClassName

Object Key Command Syntax

```
--JAR_FILE=jar_filename --MANAGER_CLASS=class
```

State Properties

Property	Value
status	ACTIVE INACTIVE

Supported Operations

```
activate  
create  
createAll  
deactivate  
delete  
deleteAll  
deleteList  
export  
exportAll  
exportList  
getAllObjectKeys  
getAllStates  
getState  
getStateList
```

Administration GUI Page

Global Settings - Identity Management Setup

XML Description

The `<search:identityPlugins>` element describes identity plug-ins:

```
<search:identityPlugins>  
  <search:identityPlugin>  
    <search:managerClassName>  
    <search:jarFilePath>
```

```

    <search:description>
    <search:version>
    <search:authAttribute>

<!-- Include parameters for activate operation -->
    <search:parameters>
      <search:parameter>
        <search:value>
        <search:description>

```

The implementation of the identity plug-in determines the parameters. You cannot create new parameters in the XML document.

Element Descriptions

<search:identityPlugins>

Contains one or more <search:identityPlugin> elements.

<search:identityPlugin>

Describes an identity plug-in. It contains these elements:

```

<search:managerClassName>
<search:jarFilePath>
<search:description>
<search:version>
<search:authAttribute>
<search:parameters>

```

<search:managerClassName>

Contains the class name of the plug-in.

<search:jarFilePath>

Contains the qualified name of the jar file. Paths can be absolute or relative to the *ORACLE_HOME*/search/lib/plugins/identity directory.

<search:description>

Contains a description of the plug-in. (Read only)

<search:version>

Contains the Oracle SES version of the plug-in. (Read only)

<search:authAttribute>

Contains the authentication attribute for the plug-in.

<search:parameters>

Contains one or more <search:parameter> elements. The parameter are used only by [activate identityPlugin](#), not by [create identityPlugin](#).

<search:parameter>

Describes a plug-in parameter. Each plug-in has its own parameters. This element contains these child elements:

```

<search:value>
<search:description>

```

Attribute	Value
name	Name of the parameter.

<search:value>

Value of the parameter.

Attribute	Value
encrypted	Indicates whether the value of <search:value> is encrypted. Set to true if the password is encrypted, or set to false if it is plain text. The default value is false.

<search:description>

Description of the parameter.

Example

This XML document defines an Oracle Internet Directory plug-in:

```
<?xml version="1.0" encoding="UTF-8" ?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:identityPlugins>
    <search:identityPlugin>
      <search:managerClassName>
        oracle.search.plugin.security.identity.oid.OIDPluginManager
      </search:managerClassName>
      <search:jarFilePath>OIDPlugins.jar</search:jarFilePath>
      <search:description>Oracle Internet Directory identity plug-in manager
      </search:description>
      <search:version>11.1.0.0.0</search:version>
      <search:authAttribute>nickname</search:authAttribute>
      <search:parameters>
        <search:parameter name="Host name">
          <search:value>my_computer</search:value>
          <search:description>OID host on my computer</search:description>
        </search:parameter>
        <search:parameter name="Port">
          <search:value>7789</search:value>
          <search:description>OID port</search:description>
        </search:parameter>
        <search:parameter name="Use SSL">
          <search:value>>false</search:value>
          <search:description>SSL encryption
          </search:description>
        </search:parameter>
        <search:parameter name="Realm">
          <search:value>dc=us,dc=example,dc=com</search:value>
          <search:description>OID realm</search:description>
        </search:parameter>
        <search:parameter name="User name">
          <search:value>cn=orcladmin</search:value>
          <search:description>OID user name</search:description>
        </search:parameter>
        <search:parameter name="Password">
          <search:value encrypted="false">mypassword</search:value>
          <search:description>Password</search:description>
        </search:parameter>
        <search:parameter name="Use User Cache">
          <search:value>>false</search:value>
          <search:description> </search:description>
        </search:parameter>
        <search:parameter name="User Cache Source Name">
```

```
        <search:description> </search:description>
    </search:parameter>
</search:parameters>
</search:identityPlugin>
</search:identityPlugins>
</search:config>
```

index

The index is a metadata repository for crawled documents and provides the search results list.

Object Type

Universal

State Properties

Property	Value
estimatedFragmentation	Decimal number representing the percent of fragmentation; optimize the index when fragmentation is greater than 50%

Supported Operations

export
getState
update

Administration GUI Page

Global Settings - Set Indexing Parameters

XML Description

The `<search:index>` element describes indexing:

```
<search:index>  
  <search:indexingBatchSize>  
  <search:indexingMemorySize>
```

Element Descriptions

<search:index>

Describes the indexing parameters. It contains these elements:

```
<search:indexingBatchSize>  
<search:indexingMemorySize>
```

<search:indexingBatchSize>

Contains the size in megabytes of the crawled documents before indexing begins. Crawling and indexing run concurrently after the initial batch size is reached. While the index is running, the crawler continues to crawl documents.

The default size is 250 MB.

<search:indexingMemorySize>

Contains the number of megabytes of memory used for indexing before swapping to disk. A large amount of memory improves both indexing and query performance.

The default size is 275 MB.

Example

This XML document configures the indexing properties:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:index>
    <search:indexingBatchSize>250</search:indexingBatchSize>
    <search:indexingMemorySize>275</search:indexingMemorySize>
  </search:index>
</search:config>
```

indexOptimizer

Optimizing the index reduces fragmentation and may significantly increase the speed of searches. In general, the fragmentation percentage should be less than 50%. A higher percentage indicates that search performance is compromised. If it is over 75%, then you should optimize the index as soon as possible.

Optimization of a very large index may take several hours. Schedule optimization during hours of low usage to ensures minimal disruption to users.

Object Type

Universal

State Properties

Property	Value
endTime	The date and time that the last optimization ended, in the form Day, DD Mon YYYY, HH:MM:SS GMT
startTime	The date and time that the last optimization started, in the same form as endTime
status	DISABLED, EXECUTING, FAILED, LAUNCHING, SCHEDULED, or STOPPED

Supported Operations

activate
deactivate
export
getState
start
update

Administration GUI Page

Global Settings - Index Optimization

XML Description

The `<search:indexOptimizer>` element describes index optimization:

```
<search:indexOptimizer>
  <search:frequency>

  <!-- For hourly optimization -->
    <search:hourly>
      <search:hoursBtwnLaunches>

  <!-- For daily optimization -->
    <search:daily>
      <search:daysBtwnLaunches>
      <search:startHour>

  <!-- For weekly optimization -->
    <search:weekly>
      <search:weeksBtwnLaunches>
```

```

        <search:startDayOfWeek>
        <search:startHour>

<!-- For monthly optimization -->
        <search:monthly>
        <search:monthsBtwnLaunches>
        <search:startDayOfMonth>
        <search:startHour>

<!-- For all frequencies -->
        <search:duration>
        <search:maxHours>

```

Element Descriptions

<search:indexOptimizer>

Describes index optimization schedule. It contains these elements:

```

<search:frequency>
<search:duration>

```

<search:frequency>

Describes the optimization schedule. It contains one of these elements:

```

<search:hourly>
<search:daily>
<search:weekly>
<search:monthly>

```

<search:hourly>

Describes an hourly schedule. It contains a <search:hoursBtwnLaunches> element.

<search:hoursBtwnLaunches>

The number of hours between optimizations.

<search:daily>

Describes a daily schedule. It contains these elements:

```

<search:daysBtwnLaunches>
<search:startHour>

```

<search:daysBtwnLaunches>

The number of days between optimizations.

<search:startHour>

The time the crawl begins using a 24-hour clock, such as 9 for 9:00 a.m. or 23 for 11:00 p.m.

<search:weekly>

Describes a weekly schedule. It contains these elements:

```

<search:weeksBtwnLaunches>
<search:startDayOfWeek>
<search:startHour>

```

<search:weeksBtwnLaunches>

The number of weeks between optimizations.

<search:startDayOfWeek>

The day of the week that the crawl begins, such as MONDAY or TUESDAY.

<search:monthly>

Describes a monthly schedule. It contains these elements:

```
<search:monthsBtwnLaunches>
<search:startDayOfMonth>
<search:startHour>
```

<search:monthsBtwnLaunches>

The number of time periods between starting a crawl.

<search:startDayOfMonth>

An integer value for the day of the month that the crawl begins, such as 1 or 15.

<search:duration>

Controls the duration of the optimization process. It contains a `<search:maxhours>` element.

Attribute	Value
haslimit	Set to true to enforce the time limit, or set to false to allow the process to finish. Required.

<search:maxHours>

The number of hours the optimization process is allowed to continue. For best results, allow the optimization to finish.

Example

This XML document contains the index optimizer settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:indexOptimizer>
    <search:frequency>
      <search:weekly>
        <search:weeksBtwnLaunches>3</search:weeksBtwnLaunches>
        <search:startDayOfWeek>MONDAY</search:startDayOfWeek>
        <search:startHour>23</search:startHour>
      </search:weekly>
    </search:frequency>
    <search:duration haslimit="true">
      <search:maxHours>8</search:maxHours>
    </search:duration>
  </search:indexOptimizer>
</search:config>
```

partitionConfig

Partitioning is used to improve the query performance of large data sets. You can use multiple partitions to distribute the document index across physical storage devices. I/O is then performed in parallel to gain the best query performance.

You can enable partitioning only on a freshly installed, empty Oracle SES instance.

See Also: "[Partitioning for Parallel Query](#)" on page 2-10

Object Type

Universal

State Properties

None

Supported Operations

export
update

Administration GUI Page

None

XML Description

The `<search:partitionConfig>` element describes partitioning:

```
<search:partitionConfig>
  <search:partitionRules>
    <search:partitionRule>
      <search:partitionValue>
      <search:valueType>
      <search:ruleType>
      <search:storageArea>
```

Element Descriptions

`<search:partitionConfig>`

Describes the partition configuration rules. It contains the `<search:partitionRules>` element.

`<search:partitionRules>`

Contains one or more `<search:partitionRule>` elements.

`<search:partitionRule>`

Describes a partition rule. It contains these elements:

```
<search:partitionValue>
<search:valueType>
<search:ruleType>
<search:ruleSetting>
<search:storageArea>
```

<search:partitionValue>

Contains the partition value, which must be EQ_DEFAULT. This metadata value represents all indexed values. Required.

<search:valueType>

Contains the type of partition value, which must be META. Required.

<search:ruleType>

Contains the type of partition rule. Required.

- HASH: Evenly distributes the index values for a large set of documents across the list of storage areas. Each partition is located in one storage area.
- VALUE: Maps the specified partition value to one partition. Oracle SES assigns this rule initially when partitioning is enabled and only one storage area is defined out of the box. Do not use this value when you plan to use multiple storage areas for parallel query. Specify HASH instead.

<search:storageArea>

A comma-separated list of storageArea objects used by this partition rule.

Example

This XML document partitions the document index across six storage areas named p1 to p6:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:partitionConfig>
    <search:partitionRules>
      <search:partitionRule>
        <search:partitionValue>EQ_DEFAULT</search:partitionValue>
        <search:valueType>META</search:valueType>
        <search:ruleType>HASH</search:ruleType>
        <search:storageArea>p1, p2, p3, p4, p5, p6</search:storageArea>
      </search:partitionRule>
    </search:partitionRules>
  </search:partitionConfig>
</search:config>
```

proxyLogin

When performing a secure search on a federation endpoint, the federation broker must transmit the identity of the user to the federation endpoint. If the endpoint instance trusts the broker instance, then the broker instance can proxy as the end user. To establish this trust relationship, Oracle SES instances exchange a secret. This secret is exchanged in the form of a trusted entity.

A trusted entity consists of two values: an entity name and an entity password. Each Oracle SES instance can have one or more trusted entities that it can use to participate in secure federated search. A federated trusted entity is also referred to as a proxy user or a proxy log-in.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Page

Global Settings - Federation Trusted Entities

XML Description

The <search:proxyLogins> element describes proxy log-ins:

```
<search:proxyLogins>
  <search:proxyLogin>
    <search:name>
    <search:password>
    <search:useIdentityPlugin>
```

Element Descriptions

<search:proxyLogins>

Describes proxy log-ins. It contains one or more <search:proxyLogin> elements.

<search:proxyLogin>

Describes a proxy log-in. It contains these elements:

```
<search:name>
<search:password>
<search:useIdentityPlugin>
```

<search:name>

Name of the proxy. Required.

<search:password>

Password for the proxy server. Required when <search:useIdentityPlugin> is false.

Attribute	Value
encrypted	Indicates whether the value of <search:password> is encrypted. Set to true if the password is encrypted, or set to false if it is plain text.

<search:useIdentityPlugin>

Controls use of an identity plug-in. Set to true to use the active identity plug-in for authentication, or set to false otherwise. Required.

Example

This XML document describes two proxy log-ins:

```
<?xml version="1.0" encoding="UTF-8" ?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:proxyLogins>
    <search:proxyLogin>
      <search:name>this_proxy</search:name>
      <search:useIdentityPlugin>true</search:useIdentityPlugin>
    </search:proxyLogin>
    <search:proxyLogin>
      <search:name>that_proxy</search:name>
      <search:password encrypted="false">password</search:password>
      <search:useIdentityPlugin>false</search:useIdentityPlugin>
    </search:proxyLogin>
  </search:proxyLogins>
</search:config>
```


queryConfig

Query configuration enables you to customize the query results and tune the search engine.

Object Type

Universal

State Properties

None

Supported Operations

export
update

Administration GUI Page

Global Settings - Query Configuration

XML Description

The `<search:queryConfig>` element sets the query configuration parameters:

```
<search:queryConfig>
  <search:maxNumResults>
  <search:displayUrls>
    <search:tableDisplayUrl>
    <search:fileDisplayUrl>
    <search:mailingListDisplayUrl>
    <search:emailDisplayUrl>

  <<search:relevancyBoosting>

  <!-- Provide spelling suggestions -->
  <search:spellingCorrection>
    <search:useLanguageDictionary>
    <search:useIndexedDocsAndQueryLog>

  <search:hitCount>
    <search:countMethod>
  <!-- For exact count methods -->
    <search:maxExactHitCount>
  <search:queryStatistics>
    <search:loggingPeriod>

  <search:urlSubmission>
  <!-- For URL Submission -->
    <search:sourceName>
    <search:checkUrlBoundaryRules>

  <search:federation>
    <search:timeout>
    <search:minNumThreads>

  <search:queryTimeAuthorization>
    <search:timeout>
```

```

<search:minNumThreads>
<search:maxNumThreads>
<search:logFilteredDocs>

<search:secureSearch>
  <search:loginRequirement>
  <search:securityFilterLifespan>
  <search:authenticationTimeout>
  <search:authorizationTimeout>
  <search:minNumThreads>
  <search:maxNumThreads>

```

Element Descriptions

<search:queryConfig>

Describes query configuration parameters. It contains these elements:

```

<search:maxNumResults>
<search:displayUrls>
<search:relevancyBoosting>
<search:spellingCorrection>
<search:hitCount>
<search:queryStatistics>
<search:urlSubmission>
<search:federation>
<search:queryTimeauthorization>
<search:secureSearch>

```

<search:maxNumResults>

Maximum number of search results returned by a query.

<search:displayUrls>

Describes the display URLs. It contains these elements:

```

<search:tableDisplayUrl>
<search:fileDisplayUrl>
<search:mailingListDisplayUrl>
<search:emailDisplayUrl>

```

<search:tableDisplayUrl>

URL used to display the retrieved data for a table source.

<search:fileDisplayUrl>

URL used to display the retrieved data for a file source.

<search:mailingListDisplayUrl>

URL used to display the retrieved data for a mailing list source.

<search:emailDisplayUrl>

URL used to display the retrieved data for an e-mail source.

<search:relevancyBoosting>

Controls relevancy boosting.

Attribute	Value
enabled	Set to true to enable relevancy boosting, or set to false otherwise. Required.

<search:spellingCorrection>

Controls spelling correction. When enabled, this element contains these child elements:

```
<search:useLanguageDictionary>
<search:useIndexedDocsAndQueryLog>
```

Attribute	Value
enabled	Set to <code>true</code> to enable spelling correction, or set to <code>false</code> otherwise. Required.

<search:useLanguageDictionary>

Controls use of an English dictionary for spelling suggestions. Set to `true` to use an English dictionary, or set to `false` to derive spelling suggestions only from terms in indexed documents and the query log. Consider `false` if users typically search for non-English terms.

<search:useIndexedDocsAndQueryLog>

Contains a value of `true` or `false` to control use of terms from indexed documents and the query log for spelling suggestions. Terms that occur frequently are extracted to the Oracle SES dictionary. Set to `true` to use terms from these sources, or set to `false` to use only the English dictionary for suggestions. Consider `false` if suggestions from crawled documents to all search users may breach security.

<search:hitCount>

Contains a `<search:countMethod>` element.

<search:countMethod>

A hit count method:

- `APPROX_COUNT`: Displays an estimated number of matching documents. This method supports better performance than `EXACT_COUNT`. (Default)
- `EXACT_COUNT`: Displays the exact number of matching documents.
- `EXACT_COUNT_QTA`: Displays the exact number of matching documents adjusted for query-time filtering.

<search:maxExactHitCount>

The maximum number of exact results. An estimated number is returned for a higher number of results.

<search:queryStatistics>

Controls the collection of search statistics. Set to `true` to collect statistics, or set to `false` otherwise. This operation degrades search performance, so you should disable it during peak hours.

Attribute	Value
enabled	Set to <code>true</code> to enable collection or set to <code>false</code> otherwise. Required.

<search:loggingPeriod>

Number of days the data is saved. Statistics are compiled for this number of days.

<search:urlSubmission>

Controls the submission of URLs. When enabled, this element contains these child elements:

```
<search:sourceName>
<search:checkUrlBoundaryRules>
```

Attribute	Value
enabled	Set to <code>true</code> to enable URL submission, or set to <code>false</code> otherwise. Required.

<search:sourceName>

A Web source to which user-suggested URLs are added.

<search:checkUrlBoundaryRules>

Controls the enforcement of boundary rules for URLs submitted by users. Set to `true` to accept only URLs that match the rules, or set to `false` to ignore the boundary rules.

<search:federation>

Describes the querying parameters of federated sources. It contains these child elements:

```
<search:timeout>
<search:minNumThreads>
<search:maxNumThreads>
```

<search:timeout>

Contains the number of milliseconds for search results to be returned.

<search:minNumThreads>

Contains the minimum number of processes to use for searching when demand is low.

<search:maxNumThreads>

Contains the maximum number of processes to use for searching when demand is high.

<search:queryTimeAuthorization>

Describes authorization. It contains these optional child elements:

```
<search:timeout>
<search:minNumThreads>
<search:maxNumThreads>
<search:logFilteredDocs>
```

<search:logFilteredDocs>

Controls document logging. Set to `true` to record all filtered documents in the query application log file, or set to `false` otherwise

Query-time filtering errors are always logged.

<search:secureSearch>

Describes secure search. It contains these child elements:

```
<search:loginRequirement>
<search:securityFilterLifespan>
<search:authenticationTimeout>
<search:authorizationTimeout>
<search:minNumThreads>
```

```
<search:maxNumThreads>
```

<search:loginRequirement>

A log-in method:

- ALL_CONTENT: Users must log in to view any content, whether public or secure.
- SECURE_CONTENT: Users must log in to view secure content.

<search:securityFilterLifespan>

Number of minutes a stored security filter is retained. Set to a value between 0 (no cache) and 526500 (one-year cache retention).

<search:authenticationTimeout>

Number of milliseconds for authentication.

<search:authorizationTimeout>

Number of milliseconds for authorization.

Example

This XML document defines the query parameters:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
<search:queryConfig>
  <search:maxNumResults>200</search:maxNumResults>
  <search:displayUrls>
    <search:tableDisplayUrl>
      /search/query/display.jsp?type=table
    </search:tableDisplayUrl>
    <search:fileDisplayUrl>
      /search/query/display.jsp?type=file
    </search:fileDisplayUrl>
    <search:mailingListDisplayUrl>
      /search/query/mail.jsp
    </search:mailingListDisplayUrl>
    <search:emailDisplayUrl>
      /search/query/pmail.jsp
    </search:emailDisplayUrl>
  </search:displayUrls>
  <search:relevancyBoosting enabled="true"/>
  <search:spellingCorrection enabled="false"/>
  <search:hitCount>
    <search:countMethod>APPROX_COUNT</search:countMethod>
  </search:hitCount>
  <search:queryStatistics enabled="true">
    <search:loggingPeriod>0</search:loggingPeriod>
  </search:queryStatistics>
  <search:urlSubmission enabled="false"/>
  <search:federation>
    <search:timeout>30000</search:timeout>
    <search:minNumThreads>5</search:minNumThreads>
    <search:maxNumThreads>20</search:maxNumThreads>
  </search:federation>
  <search:queryTimeAuthorization>
    <search:timeout>30000</search:timeout>
    <search:minNumThreads>5</search:minNumThreads>
    <search:maxNumThreads>20</search:maxNumThreads>
    <search:logFilteredDocs>>false</search:logFilteredDocs>
```

```
</search:queryTimeAuthorization>
<search:secureSearch>
  <search:loginRequirement>SECURE_CONTENT</search:loginRequirement>
  <search:securityFilterLifespan>60</search:securityFilterLifespan>
  <search:authenticationTimeout>10000</search:authenticationTimeout>
  <search:authorizationTimeout>10000</search:authorizationTimeout>
  <search:minNumThreads>5</search:minNumThreads>
  <search:maxNumThreads>20</search:maxNumThreads>
</search:secureSearch>
</search:queryConfig>
```

resultList

The result list settings enable you to select the attributes included in the search results and customize the look-and-feel of the Oracle SES Search Application.

Object Type

Universal

State Properties

Property	Value
status	ACTIVE
	INACTIVE

Supported Operations

activate
deactivate
export
getState
update

Administration GUI Page

Global Settings - Configure Search Result List

XML Description

The `<search:resultList>` element describes the search results lists:

```
<search:resultList>
  <search:renderingAttrs>
    <search:renderingAttr>
      <search:name>
    <search:xsltContent>
  <search:cssContent>
```

Element Descriptions

`<search:resultList>`

Contains these elements:

```
<search:renderingAttrs>
<search:xsltContent>
<search:cssContent>
```

`<search:renderingAttrs>`

Identifies attributes that appear in the search results, including local search attributes, federated search attributes, and Oracle SES internal attributes. It contains one or more `<search:renderingAttr>` elements.

`<search:renderingAttr>`

Identifies an attribute. It contains a `<search:name>` element.

<search:name>

Contains the name of an attribute. Required.

<search:xsltContent>

Contains the content of an XSLT style sheet in XML-escaped format or wrapped in a CDATA element. The XSLT operates on the attributes by transforming the XML content into an HTML fragment for display in the result list. To return HTML, include this in the XSLT:

```
<xsl:output method="html" />
```

If the XSLT is blank, then the search results are displayed as untransformed XML.

<search:cssContent>

Content of a cascading style sheet (CSS) wrapped in a CDATA element. These styles format the HTML returned by the XSLT style sheet.

This CSS is used with other style sheets installed with the Oracle SES and has the highest priority.

Example

This XML document contains the result list properties and style sheets:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion=".*" xmlns:search="http://xmlns.oracle.com/search">
  <search:resultList>
    <search:renderingAttrs>
      <search:renderingAttr>
        <search:name>Subject</search:name>
      </search:renderingAttr>
      <search:renderingAttr>
        <search:name>eqdatasourcename</search:name>
      </search:renderingAttr>
      <search:renderingAttr>
        <search:name>eqdatasourcetype</search:name>
      </search:renderingAttr>
    </search:renderingAttrs>
    <search:xsltContent>
      <![CDATA[<?xml version="1.0" encoding="UTF-8" ?>
        <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

          <!-- XSLT content appears here-->

        </xsl:stylesheet>]]>
    </search:xsltContent>
    <search:cssContent>
      <![CDATA[.title
      {
        font-size: 12pt;
      }]]>
    </search:cssContent>
  </search:resultList>
</search:config>
```

schedule

Schedules define the frequency of updating the index with information about each source.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

Property	Value
lastCrawled	The date of the last scheduled crawl in the format Day, DD MMM YYYY HH:MM:SS GMT
logFilePath	The full path to the crawler log files
nextCrawl	The date of the next scheduled crawl in the same format as lastCrawled.
scheduleError	The text of the last error message
status	DISABLED, EXECUTING, FAILED, LAUNCHING, PARTIALLY_FAILED, SCHEDULED, or STOPPED

Supported Operations

activate
 create
 createAll
 deactivate
 delete
 deleteAll
 deleteList
 export
 exportAll
 exportList
 getAllObjectKeys
 getAllStates
 getState
 getStateList
 start
 stop
 update
 updateAll

Administration GUI Page

Home - Schedules - Create or Edit Schedule

XML Description

A `<search:schedules>` element describes the schedules for crawling sources:

```

<search:schedules>
  <search:schedule>
    <search:name>
    <search:crawlingMode>
    <search:recrawlPolicy>
    <search:frequency>

<!-- For hourly crawls: -->
  <search:hourly>
    <search:hoursBtwnLaunches>

<!-- For daily crawls: -->
  <search:daily>
    <search:daysBtwnLaunches>
    <search:startHour>

<!-- For weekly crawls: -->
  <search:weekly>
    <search:weeksBtwnLaunches>
    <search:startDayOfWeek>
    <search:startHour>

<!-- For monthly crawls: -->
  <search:monthly>
    <search:monthsBtwnLaunches>
    <search:startDayOfMonth>
    <search:startHour>

<!-- For manual crawls: -->
  <search>manual>

<!-- For all crawls: -->
  <search:assignedSources>
    <search:assignedSource>

```

Element Descriptions

`<search:schedules>`

Contains one or more `<search:schedule>` elements, one for each schedule.

`<search:schedule>`

Describes a schedule for crawling sources. It contains these elements:

```

<search:name>
<search:crawlingMode>
<search:recrawlPolicy>
<search:frequency>
<search:assignedSources>

```

`<search:name>`

The name of the schedule. Required.

`<search:crawlingMode>`

A crawling mode:

- **ACCEPT_ALL**: Crawls and indexes all URLs in the source, and extracts and indexes any links found in the URLs of Web sources. If the URL has been crawled before, then it is reindexed only after it changes.
- **EXAMINE_URLS**: Crawls but does not index any URLs in the source. It also crawls any links found in those URLs. Use this mode when first crawling a new source, so that you can examine the documents and refine the crawling parameters if necessary before indexing.
- **INDEX_ONLY**: Crawls and indexes all URLs in the source. It does not extract any links from those URLs. In general, select this option for a source that has been crawled previously using **EXAMINE_URLS**.

<search:recrawlPolicy>

A recrawl policy:

- **PROCESS_ALL**: Recrawls all documents in the source.
- **PROCESS_CHANGED**: Crawls only documents that changed after the last crawl. For file sources, documents are also crawled if the parent directory changed.

<search:frequency>

Controls the intervals between starting a schedule. It contains one of these elements:

```
<search:hourly>
<search:daily>
<search:weekly>
<search:monthly>
<search>manual>
```

<search:hourly>

Describes an hourly schedule. It contains a `<search:hoursBtwnLaunches>` element.

<search:hoursBtwnLaunches>

Number of hours between starting crawls, in the range of 1 to 23.

<search:daily>

Describes a daily schedule. It contains these elements:

```
<search:daysBtwnLaunches>
<search:startHour>
```

<search:daysBtwnLaunches>

Number of days between starting crawls, in the range of 1 to 99.

<search:startHour>

The time the crawl begins using a 24-hour clock, such as 9 for 9:00 a.m. or 23 for 11:00 p.m.

<search:weekly>

Describes a weekly schedule. It contains these elements:

```
<search:weeksBtwnLaunches>
<search:startDayOfWeek>
<search:startHour>
```

<search:weeksBtwnLaunches>

Number of weeks between starting crawls, in the range of 1 to 12.

<search:startDayOfWeek>

The day of the week that the crawl begins, such as MONDAY or TUESDAY.

<search:monthly>

Describes a monthly schedule. It contains these elements:

```
<search:monthsBtwnLaunches>
<search:startDayOfMonth>
<search:startHour>
```

<search:monthsBtwnLaunches>

Number of months between starting crawls, in the range of 1 to 12.

<search:startDayOfMonth>

An integer value for the day of the month that the crawl begins, such as 1 or 15.

<search>manual>

Describes a manual search.

<search:assignedSources>

Contains one or more <search:assignedSource> elements, one for each source that is crawled using this schedule.

<search:assignedSource>

The name of a source crawled using this schedule. The source cannot be a mailing-list source or a federated source.

Example

This XML document creates a schedule for mySource that runs every third Monday at 11:00 p.m.:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:schedules>
    <search:schedule>
      <search:name>schedule1</search:name>
      <search:crawlingMode>INDEX_ONLY</search:crawlingMode>
      <search:recrawlPolicy>PROCESS_ALL</search:recrawlPolicy>
      <search:frequency>
        <search:weekly>
          <search:weeksBtwnLaunches>3</search:weeksBtwnLaunches>
          <search:startDayOfWeek>MONDAY</search:startDayOfWeek>
          <search:startHour>23</search:startHour>
        </search:weekly>
      </search:frequency>
      <search:assignedSources>
        <search:assignedSource>mySource</search:assignedSource>
      </search:assignedSources>
    </search:schedule>
  </search:schedules>
</search:config>
```

searchAttr

Search attributes are attributes exposed to the search user. Oracle Secure Enterprise Search (SES) provides system-defined attributes, such as author and description, and enables administrators to create custom attributes.

When the indexed documents contain metadata, such as author and date information, you can let users refine their searches based on this information. For example, users can search for all documents by a particular author, that is, where the author attribute has a particular value.

Oracle Secure Enterprise Search has several default search attributes. They can be incorporated in search applications for a more detailed search and richer presentation. If an attribute List of Values (LOV) is available, then the crawler registers the LOV definition, which includes attribute value, attribute value display name, and its translation.

You can create, delete, and update custom attributes, and update the default attributes.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Pages

Global Settings - Search Attributes
Global Settings - Search Attributes - Manage LOVs for Attribute
Global Settings - Translate Search Attribute Name
Global Settings - Translate LOV Display Name

XML Description

The `<search:Attrs>` element describes search attributes:

```
<search:searchAttrs>
  <search:searchAttr>
    <search:name>
    <search:type>
    <search:translations>
    <search:lovEntries>
      <search:lovEntry>>
        <search:lovValue>
        <search:sourceName>
        <search:translations>
```

Element Descriptions

<search:searchAttrs>

Contains one or more `<search:searchAttr>` elements.

<search:searchAttr>

Describes a search attribute. It contains these elements:

```
<search:name>
<search:type>
<search:translations>
<search:lovEntries>
```

<search:name>

Name of the search attribute. (Required)

<search:type>

Data type of the attribute values. Set to `STRING`, `NUMBER`, or `DATE`. (Required)

<search:translations>

Provides a display name. See ["Providing Translations of Object Names"](#) on page 2-6.

<search:lovEntries>

Contains one or more `<search:lovEntry>` elements, each describing a list of values (LOV).

```
<search:lovValue>
<search:sourceName>
<search:translations>
```

<search:lovEntry>

Describes a list of values. It contains these child elements:

<search:lovValue>

Name of the list of values. (Required)

<search:sourceName>

Name of the source for a source-specific list of values.

Example

This XML document defines a search attribute named `Copyright`:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
```

```
<search:searchAttrs>
  <search:searchAttr>
    <search:name>Copyright</search:name>
    <search:type>DATE</search:type>
  </search:searchAttr>
</search:searchAttrs>
</search:config>
```

skinBundle

A skin bundle is a set of files that customize the look and feel of the Oracle SES default query application.

See Also: ["Search Interface Customization: Skin Bundles"](#) on page 2-13

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

Property	Value
status	ACTIVE or INACTIVE

Supported Operations

activate
create
deactivate
delete
deleteAll
deleteList
export
exportAll
exportList
getAllStates
getState
getStateList
update

Administration GUI Page

None

XML Description

The `<search:skinBundles>` element describes skin bundles:

```
<search:skinBundles>  
  <search:skinBundle>  
    <search:name>  
    <search:isDefault>  
    <search:linkedVersion>  
    <search:files>  
      <search:file>
```


Element Descriptions

<search:skinBundles>

Contains one or more <search:skinBundle> elements.

<search:skinBundle>

Describes a skin bundle. It contains these elements:

```
<search:name>
<search:isDefault>
<search:linkedVersion>
<search:files>
```

<search:name>

Contains the name of the skin bundle. (Required)

<search:isDefault>

Identifies whether this is the default skin bundle. Set to `true` to make this the default skin bundle; otherwise, set it to `false`.

<search:linkedVersion>

Contains the version number of Oracle SES.

<search:files>

Contains one or more <search:file> elements.

<search:file>

Identifies the path to a file composing the skin bundle, such as a template (ftl), cascading style sheet (css), JavaScript (js), or graphic (gif).

Attribute	Value
path	Relative path of the file in the skin bundle. (Required)

Examples

This example describes a skin bundle named `acme`.

```
<?xml version="1.0" encoding="UTF-8" ?>

<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:skinBundles>
    <search:skinBundle>
      <search:name>acme</search:name>
      <search:isDefault>false</search:isDefault>
      <search:linkedVersion>11.1.2.2.0</search:linkedVersion>
      <search:files>
        <search:file path="templates/query.ftl"/>
        <search:file path="templates/inc_footer.ftl"/>
        <search:file path="assets/images/logo.gif"/>
        <search:file path="assets/css/acme.css"/>
      </search:files>
    </search:skinBundle>
  </search:skinBundles>
</search:config>
```

SOURCE

Sources are collections of data to be searched, such as Web sites, files, database tables, content management repositories, collaboration repositories, and applications.

Note: The current release of the Oracle SES Administration API supports these source types:

- File
 - Federated
 - User Defined
 - Web
-
-

Object Type

Creatable

Object Key

name

Object Key Command Syntax

`--NAME=object_name`

`-n object_name`

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Page

Home - Sources - Create or Edit Source
Home - Sources - Customize Federated Source

XML Descriptions

Each supported source type has a unique XML description:

- [XML Description: Federated Sources](#)

- XML Description: File Sources
- XML Description: User-Defined Sources
- XML Description: Web Sources

XML Description: Federated Sources

For a federated source, the `<search:sources>` element contains a `<search:federatedSource>` element:

```

<search:sources>
  <search:federatedSource>
    <search:name>
    <search:url>
    <search:security>
      <search:entityName>
      <search:entityPassword>
      <search:authAttribute>
    <search:queryRouting>
    <search:filterRule>
    <search:searchRestrictions>
      <search:groupRestrictedEnabled>
      <search:searchedGroups>
        <search:fedSourceGroup>
    <search:attributeRetrieval>
      <search:retrievedAttrs>
        <search:fedSearchAttr>
      <search:attributeMappings>
        <search:attributeMapping>
          <search:localAttribute>
          <search:localAttribute>

```

Element Descriptions

`<search:sources>`

Contains one or more source descriptions.

`<search:federatedSource>`

Describes a federated source. It contains these elements:

```

<search:name>
<search:url>
<search:security>
<search:queryRouting>
<search:searchRestrictions>
<search:attributeRetrieval>

```

`<search:name>`

Contains the name of the source. (Required)

`<search:url>`

Contains the Web service URL.

`<search:security>`

Describes security for connecting to the federated source. It contains these child elements:

```

<search:entityName>
<search:entityPassword>
<search:authAttribute>

```

<search:entityName>

Contains the name of the federation trusted entity on the federation endpoint. Contact the administrator of the federated endpoint for this information

<search:entityPassword>

Contains the password for the entity name.

Attribute	Value
encrypted	Indicates whether the value of <search:entityPassword> is encrypted. Set to <code>true</code> if the password is encrypted, or set to <code>false</code> if it is plain text.

<search:authAttribute>

Contains the name of an attribute that identifies and can authenticate a user on the federation endpoint.

<search:queryRouting>

Describes the rules for routing queries to the federated source. Without any rules, Oracle SES routes all queries to the federated source. This element is optional, but can improve scalability. It contains a <search:filterRule> element.

<search:filterRule>

Contains the rules within a CDATA element. Rules consist of an attribute, a colon (:), and an expression. Attributes can be DATE, STRING, or NUMBER. DATE and NUMBER attributes can include these operators: -, =, >, >=, <, <=. The AND or OR operators separate multiple rules.

<search:searchRestrictions>

Restricts searches to a list of source groups. It contains these child elements:

```
<search:groupRestrictedEnabled>
<search:searchedGroups>
```

<search:groupRestrictedEnabled>

Controls whether source groups are restricted during searches. Set to `true` to restrict searches, or set to `false` otherwise. The default value is `false`. (Optional)

<search:searchedGroups>

Describes the source groups to be searched on the federated source. It contains one or more <search:fedSourceGroup> elements.

<search:fedSourceGroup>

Empty element that uses parameters to identify source group. (Read only)

Attribute	Value
isAvailable	Identifies whether the source group is currently available in the federated source.
name	Name of a federated source group. (Required)

<search:attributeRetrieval>

Describes the attributes to be retrieved from the federated source. It contains a <search:retrieveAttrs> element.

<search:retrievedAttrs>

Contains one or more <search:fedSearchAttr> elements.

<search:fedSearchAttr>

Empty element that uses parameters to identify a search attribute.

Attribute	Value
name	Name of a search attribute. (Required)
type	Data type of the attribute: <code>STRING</code> , <code>NUMBER</code> , or <code>DATE</code> .
isAvailable	Identifies whether the attribute is currently available in the federated source: <code>true</code> if it is available, or <code>false</code> otherwise.
isMandatory	Identifies whether retrieval of the attribute is mandatory: <code>true</code> if it must be listed in the <code><search:retrievedAttrs></code> element, or <code>false</code> if it can be omitted without causing an error.

<search:attributeMappings>

Contains one or more `<search:attributeMapping>` elements.

<search:attributeMapping>

Maps a local attribute to a remote attribute. It contains one of each of these elements:

```
<search:localAttribute>
<search:remoteAttribute>
```

<search:localAttribute>

Identifies the local attribute being mapped.

Attribute	Value
name	Name of the local attribute. (Required)
type	Data type of the local attribute: <code>STRING</code> , <code>NUMBER</code> , or <code>DATE</code> . (Required)

<search:remoteAttribute>

Identifies the remote attribute being mapped.

Attribute	Value
name	Name of the remote attribute. (Required)
type	Data type of the remote attribute: <code>STRING</code> , <code>NUMBER</code> , or <code>DATE</code> . (Required)
isAvailable	Identifies whether the remote attribute is currently available in the federated source: <code>true</code> if it is available, or <code>false</code> otherwise.

Example 2-1 Federated Source Description

This XML document describes a federated source:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sources>
    <search:federatedSource>
      <search:name>fed1</search:name>
      <search:url>http://example:7777/search/query/OracleSearch</search:url>
      <search:security>
        <search:entityName>entity2</search:entityName>
        <search:entityPassword encrypted="false">password</search:entityPassword>
        <search:authAttribute>nickname</search:authAttribute>
      </search:security>
    </search:federatedSource>
  </search:sources>
</search:config>
```

```

</search:security>
<search:queryRouting>
  <search:filterRule>
    <![CDATA[
      (language:en) AND (idm::mail:a.*)
    ]]>
  </search:filterRule>
</search:queryRouting>
<search:searchRestrictions>
  <search:groupRestrictedEnabled>true</search:groupRestrictedEnabled>
  <search:searchedGroups>
    <search:fedSourceGroup isAvailable="true" name="FILE"/>
    <search:fedSourceGroup isAvailable="true" name="Web"/>
  </search:searchedGroups>
</search:searchRestrictions>
<search:attributeRetrieval>
  <search:retrievedAttrs>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Author"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Description"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Infosource"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Infosource Path"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Language"/>
    <search:fedSearchAttr type="DATE" isAvailable="true"
      isMandatory="true" name="LastModifiedDate"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Mimetype"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Title"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="Url"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="false" name="custom1"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="false" name="custom2"/>
    <search:fedSearchAttr type="NUMBER" isAvailable="true"
      isMandatory="true" name="eqdocid"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="eqfedid"/>
    <search:fedSearchAttr type="STRING" isAvailable="true"
      isMandatory="true" name="eqsnippet"/>
  </search:retrievedAttrs>
</search:attributeRetrieval>
</search:federatedSource>
</search:sources>
</search:config>

```

XML Description: File Sources

For a file source, the `<search:sources>` element contains a `<search:fileSource>` element:

```

<search:sources>
  <search:fileSource>
    <search:name>
    <search:fileDisplayUrl>
    <search:fileUrlPrefix>

```

```

    <search:displayUrlPrefix>
  <search:startingUrls>
    <search:startingUrl>
      <search:url>
    <search:aclPolicy>
  <search:authorizationPlugin>
  <search:boundaryRules>
  <search:attributeMappings>
    <search:attributeMapping>
      <search:documentAttr>
      <search:searchAttr>
  <search:crawlerSettings>
    <search:numThreads>
    <search:languageDetection>
    <search:defaultLanguage>
    <search:crawlTimeout>
    <search:maxDocumentSize>
    <search:preserveDocumentCache>
    <search:defaultCharSet>
    <search:servicePipeline>
      <search:pipelineName>
  <search:documentTypes>
    <search:documentType>
    <search:mimeType>

```

Element Descriptions

<search:sources>

Contains one or more source descriptions.

<search:fileSource>

Describes a file source. It contains these elements:

```

<search:name>
<search:fileDisplayUrl>
<search:startingUrls>
<search:aclPolicy>
<search:boundaryRules>
<search:attributeMappings>
<search:crawlerSettings>
<search:documentTypes>

```

<search:name>

Contains the name of the file source.

<search:fileDisplayUrl>

Identifies a physical path that is replaced by a display URL for security reasons when the file is retrieved during a search.

Attribute	Value
enabled	Controls whether the display URL prefix is used for security reasons. Set to <code>true</code> to use the display URL, or set to <code>false</code> to display the physical location of the file. (Required)

<search:fileUrlPrefix>

Contains the physical file URL to be replaced by the display URL.

<search:displayUrlPrefix>

Contains a URL prefix displayed instead of the file URL.

<search:startingUrls>

Identifies the file path where the crawler begins. It consists of one or more of these child elements:

<search:startingUrl>

Contains a `<search:url>` element.

<search:url>

Contains an entry point for starting to crawl files. The URL must be in its original form as an unencoded file path.

<search:aclPolicy>

Describes an authorization policy for the source. See ["XML Description: Web Sources"](#) on page 2-81.

<search:authorizationPlugin>

Describes the authorization plug-in. See ["XML Description: User-Defined Sources"](#) on page 2-77.

<search:boundaryRules>

Describes the boundary rules for the source. See ["XML Description: Web Sources"](#) on page 2-81.

<search:attributeMappings>

Maps the document attributes to search attributes. It contains one or more `<search:attributeMapping>` elements.

<search:attributeMapping>

Contains a document attribute and a search attribute for mapping. It contains one of each of these child elements:

`<search:documentAttr>`

`<search:searchAttr>`

<search:documentAttr>

Identifies a document attribute by its name and data type.

Attribute	Value
name	Name of a document attribute
type	Data type of the attribute: DATE, NUMBER, or STRING

<search:searchAttr>

Identifies a search attribute by its name and data type. Search attributes are displayed to users in the Oracle SES Search interface.

Attribute	Value
name	Name of a search attribute
type	Data type of the attribute: DATE, NUMBER, or STRING

<search:crawlerSettings>

Configures the crawler. It contains these child elements:


```

<search:numThreads>
<search:languageDetection>
<search:defaultLanguage>
<search:crawlTimeout>
<search:maxDocumentSize>
<search:preserveDocumentCache>
<search:defaultCharSet>
<search:servicePipeline>

```

<search:numThreads>

Contains the number of simultaneous processes available for crawling.

<search:languageDetection>

Controls the use of a language detector when the metadata for a document does not identify the language.

Attribute	value
enabled	Controls use of language detection when a source document does not indicate the language in the header. Set to <code>true</code> to enable language detection, or set to <code>false</code> otherwise. (Required)

<search:defaultLanguage>

Default language used by the crawler when the document language is not identified.

<search:crawlTimeout>

Contains the number of milliseconds allowed for the target site to return a document.

<search:maxDocumentSize>

Contains the maximum document size in megabytes. Larger documents are not crawled.

<search:preserveDocumentCache>

Controls retention of the document cache after indexing.

Attribute	Value
enabled	Set to <code>true</code> to retain the cache, or set to <code>false</code> otherwise. (Required)

<search:defaultCharSet>

Code for the default character set, which is used when a source document does not identify its character set in the metadata

<search:servicePipeline>

Controls use of a document service pipeline. When enabled, this element contains a `<search:pipelineName>` element.

Attribute	Value
enabled	Set to <code>true</code> to use the pipeline, or set to <code>false</code> otherwise. (Required)

<search:pipelineName>

Contains the name of the pipeline.

<search:documentTypes>

Identifies the types of documents to be crawled. It contains one or more `<search:documentType>` elements.

<search:documentType>

Contains one or more <search:mimeType> elements.

<search:mimeType>

Contains the Internet media type of the content in the form *type/subtype*. See [Table 2–1, "Document Formats"](#) for supported MIME types.

Example 2–2 File Source Description

This XML document describes a file source:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sources>
    <search:fileSource>
      <search:name>Document Library</search:name>
      <search:fileDisplayUrl enabled="false"/>
      <search:startingUrls>
        <search:startingUrl>
          <search:url>file://localhost/startingDirectory/</search:url>
        </search:startingUrl>
      </search:startingUrls>
      <search:aclPolicy>
        <search:noACL/>
      </search:aclPolicy>
      <search:attributeMappings>
        <search:attributeMapping>
          <search:documentAttr name="AUTHOR" type="STRING"/>
          <search:searchAttr name="Author" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="CREATOR" type="STRING"/>
          <search:searchAttr name="Author" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="DESCRIPTION" type="STRING"/>
          <search:searchAttr name="Description" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="HOST" type="STRING"/>
          <search:searchAttr name="Host" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="INFOSOURCE" type="STRING"/>
          <search:searchAttr name="Infosource" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="KEYWORD" type="STRING"/>
          <search:searchAttr name="Keywords" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="KEYWORDS" type="STRING"/>
          <search:searchAttr name="Keywords" type="STRING"/>
        </search:attributeMapping>
        <search:attributeMapping>
          <search:documentAttr name="LANGUAGE" type="STRING"/>
          <search:searchAttr name="Language" type="STRING"/>
        </search:attributeMapping>
      </search:attributeMappings>
    </search:fileSource>
  </search:sources>
</search:config>
```

```

        <search:documentAttr name="LASTMODIFIEDDATE" type="DATE"/>
        <search:searchAttr name="LastModifiedDate" type="DATE"/>
    </search:attributeMapping>
</search:attributeMapping>
    <search:documentAttr name="MIMETYPE" type="STRING"/>
    <search:searchAttr name="Mimetype" type="STRING"/>
</search:attributeMapping>
</search:attributeMapping>
    <search:documentAttr name="SUBJECT" type="STRING"/>
    <search:searchAttr name="Subject" type="STRING"/>
</search:attributeMapping>
</search:attributeMapping>
    <search:documentAttr name="SUBJECTS" type="STRING"/>
    <search:searchAttr name="Subject" type="STRING"/>
</search:attributeMapping>
</search:attributeMapping>
    <search:documentAttr name="TITLE" type="STRING"/>
    <search:searchAttr name="Title" type="STRING"/>
</search:attributeMapping>
</search:attributeMappings>
<search:crawlerSettings>
    <search:numThreads>5</search:numThreads>
    <search:languageDetection enabled="false"/>
    <search:defaultLanguage>en</search:defaultLanguage>
    <search:crawlTimeout>30</search:crawlTimeout>
    <search:maxDocumentSize>10</search:maxDocumentSize>
    <search:preserveDocumentCache enabled="true"/>
    <search:defaultCharSet>8859_1</search:defaultCharSet>
    <search:servicePipeline enabled="true">
        <search:pipelineName>Default pipeline</search:pipelineName>
    </search:servicePipeline>
</search:crawlerSettings>
<search:documentTypes>
    <search:documentType>
        <search:mimeType>text/html</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>text/plain</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>text/xml</search:mimeType>
    </search:documentType>
</search:documentTypes>
</search:fileSource>
</search:sources>
</search:config>

```

XML Description: User-Defined Sources

For a user-defined source, a `<search:sources>` element contains a `<search:userDefinedSource>` element:

```

<search:sources>
  <search:userDefinedSource>
    <search:name>
    <search:sourceTypeName>
    <search:aclPolicy>
    <search:authorizationPlugin>
      <search:managerClassName>
      <search:jarFilePath>
      <search:parameters>

```

```

    <search:parameter>
  <search:securityAttrs>
    <search:securityAttr>
  <search:parameters>
    <search:parameter>
      <search:value>
    <search:boundaryRules>
  <search:attributeMappings>
  <search:crawlerSettings>
  <search:documentTypes>
    <search:documentType>
    <search:mimeType>

```

Element Descriptions

<search:sources>

Describes one or more sources.

<search:userDefinedSource>

Describes a user-defined source. It contains these child elements:

```

<search:name>
<search:sourceTypeName>
<search:boundaryRules>
<search:aclPolicy>
<search:attributeMappings>
<search:documentTypes>
<search:parameters>

```

<search:name>

Name of the user-defined source.

<search:sourceTypeName>

Type of user-defined source. For a complete list of user-defined source types, issue an `exportAll sourceType` command. Set to the source type exactly as shown.

```

Database
EMC Documentum Content Server
EMC Documentum eRoom
Federated User Authorization Cache
FileNet Content Engine 1
FileNet Image Services 1
Hummingbird 1
IBM DB2 1
Lotus Notes
Microsoft Exchange)
Microsoft SharePoint 2007
NTFS
Open Text Livelink 1
Oracle Calendar
Oracle Collaboration Suite E-Mail
Oracle Content Database
Oracle Content Database (JDBC)
Oracle Content Server
Oracle E-Business Suite
Oracle Fusion

```

¹ These connectors are deprecated in this release, so that maintenance will be provided only when issues are raised by existing customers. No active development, testing, or certification will be provided for any of these connectors.

Oracle WebCenter
 Siebel 7.8
 Siebel 7.8(Public)
 Siebel 8
 User Authorization Cache
User-Defined Source Type

<search:aclPolicy>

See ["XML Description: Web Sources"](#) on page 2-81.

<search:authorizationPlugin>

Describes an authorization plug-in. It contains these elements:

```
<search:managerClassName>
<search:jarFilePath>
<search:parameters>
```

<search:managerClassName>

Contains the name of the plug-in manager Java class.

<search:jarFilePath>

Contains the qualified name of the jar file. Paths can be absolute or relative to the *ORACLE_HOME*/search/lib/plugins/identity directory.

<search:parameters>

Contains one or more `<search:parameter>` elements, each one setting a parameter. This element appears in a `<search:userDefinedSource>` element to define parameters supported by the source. It also appears in a `<search:authorizationPlugin>` to define parameters supported by the plug-in.

<search:parameter>

Describes a parameter. It contains the following elements:

```
<search:value>
<search:description>
```

Attribute	Value
name	Name of a parameter.

<search:value>

Contains the value of the parameter.

Attribute	Value
encrypted	Indicates whether the value of <code><search:value></code> is encrypted. Set to <code>true</code> if the value is encrypted, or set to <code>false</code> if it is plain text.

<search:description>

Contains a description of the parameter.

<search:securityAttrs>

Contains one or more `<search:securityAttr>` elements.

<search:securityAttr>

Contains a user or a group that is granted or denies access to the data source, depending on the value of the type attribute. (Read only)

Attribute	Value
type	Set to GRANT if the user or group has access to the source, or set to DENY otherwise.

<search:boundaryRules>

Describes the boundary rules. See ["XML Description: Web Sources"](#) on page 2-81.

<search:attributeMappings>

Maps the document attributes to search attributes. See ["XML Description: File Sources"](#) on page 2-72.

<search:crawlerSettings>

Configures the crawler. It contains these child elements:

```
<search:numThreads>
<search:languageDetection>
<search:defaultLanguage>
<search:crawlTimeout>
<search:maxDocumentSize>
<search:preserveDocumentCache>
<search:defaultCharSet>
<search:servicePipeline>
```

See ["XML Description: Web Sources"](#) on page 2-81.

<search:documentTypes>

Identifies the types of documents to be crawled. It contains one or more <search:documentType> elements.

<search:documentType>

Contains a <search:mimeType> element.

<search:mimeType>

Contains the Internet media type of the content in the form *type/subtype*. See [Table 2-1, "Document Formats"](#).

Example 2-3 User-Defined Source Description

This XML document describes an Oracle Content Database source.

```
<?xml version="1.0"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sources>
    <search:userDefinedSource>
      <search:name>contentdb</search:name>
      <search:sourceTypeName>Oracle Content Database</search:sourceTypeName>
      <search:aclPolicy>
        <search:noACL/>
      </search:aclPolicy>
      <search:parameters>
        <search:parameter name="Oracle Content Database URL">
          <search:value>http://contentDBUrl.com:7777/content</search:value>
        </search:parameter>
        <search:parameter name="Starting paths">
          <search:value>/us</search:value>
        </search:parameter>
        <search:parameter name="Depth">
```

```

        <search:value>-1</search:value>
    </search:parameter>
    <search:parameter name="Oracle Content Database admin user">
        <search:value>myUserName</search:value>
    </search:parameter>
    <search:parameter name="Entity name">
        <search:value>
        orclapplicationcommonname=ocscsplugin,cn=ifs,cn=products,cn=oraclecontext
        </search:value>
    </search:parameter>
    <search:parameter name="Entity password">
        <search:value encrypted="false">password</search:value>
    </search:parameter>
    <search:parameter name="Crawl only">
        <search:value>>false</search:value>
    </search:parameter>
    <search:parameter name="Use e-mail for authorization">
        <search:value>>false</search:value>
    </search:parameter>
</search:parameters>
</search:userDefinedSource>
</search:sources>
</search:config>

```

XML Description: Web Sources

For a Web source, the `<search:source>` element contains a `<search:webSource>` element:

```

<search:sources>
  <search:webSource>
    <search:name>
    <search:selfService>
    <search:startingUrls>
      <search:startingUrl>
        <search:url>

        <search:aclPolicy>
<!-- No ACL policy -->
        <search:noACL>
<!-- Document-level ACL policy -->
        <search:documentLevelACL>
<!-- Source-level ACL policy -->
        <search:sourceLevelACL>
          <search:accessControlEntries>
            <search:accessControlEntry>
              <search:name>
              <search:privilege>

          <search:authorizationPlugin>
<!-- Boundary rules -->
          <search:boundaryRules>
            <search:boundaryRule>
              <search:ruleType>
              <search:ruleOperation>
              <search:rulePattern>

          <search:metatagMappings>
            <search:metatagMapping>
              <search:documentAttr>
              <search:searchAttr>

```

```
<search:crawlerSettings>
  <search:numThreads>
  <search:languageDetection>
    <search:defaultLanguage>
  <search:crawlDepth>
    <search:limit>
  <search:crawlTimeout>
  <search:maxDocumentSize>
  <search:preserveDocumentCache>
  <search:defaultCharSet>
  <search:servicePipeline>
    <search:pipelineName>
  <search:honorRobotsExclusion>
  <search:indexDynamicPages>
  <search:urlRewriter>
    <search:urlRewriterClass>
    <search:urlRewriterJar>
  <search:httpCharSetOverride>
  <search:cookies>
    <search:cookieContentInLog>
    <search:maxCookieSize>
    <search:maxCookies>
    <search:maxCookiesPerHost>

<search:documentTypes>
  <search:documentType>
  <search:mimeType>

<search:httpAuthentications>
  <search:httpAuthentication>
    <search:host>
    <search:realm>
    <search:username>
    <search:password>

<search:htmlForms>
  <search:htmlForm>
    <search:name>
    <search:formUrl>
    <search:action>
    <search:successUrl>
    <search:formControls>
      <search:formControl>
        <search:name>
        <search:value>
        <search:isPasswordField>

<search:ssoAuthentication>
  <search:username>
  <search:password>
```

Element Descriptions

<search:sources>

Contains one or more source descriptions.

<search:webSource>

Describes a Web source. It contains these child elements:


```

<search:name>
<search:selfService
<search:startingUrls>
<search:aclPolicy>
<search:boundaryRules>
<search:metatagMappings>
<search:crawlerSettings>
<search:documentTypes>
<search:httpAuthentications>
<search:htmlForms>
<search:ssoAuthentication>

```

<search:name>

Name of the Web source.

<search:selfService>

Contains a value of `true` to enable self-service authentication, or a value of `false` to disable it. Self-service authentication lets users enter authentication credentials at run time, instead of the administrator entering credentials at the time the source is created.

<search:startingUrls>

Contains one or more `<search:startingUrl>` elements.

<search:startingUrl>

Contains a `<search:url>` element.

<search:url>

Contains the URL-encoded Web address that is an entry point for starting to crawl Web pages.

<search:aclPolicy>

Describes an ACL policy for the source. It contains one of these child elements:

```

<search:noACL>
<search:documentLevelACL>
<search:sourceLevelACL>

```

<search:noACL>

Indicates no ACL policy. All documents are visible and searchable.

<search:documentLevelACL>

Describes a document-level ACL policy.

<search:sourceLevelACL>

Describes an Oracle SES ACL policy used when crawling private content. It preserves authorizations specified in OracleAS Portal. For user-defined sources, crawler plug-ins (or connectors) can supply ACL information with documents for indexing, which provides finer control document protection. That is, each document within one source may be viewed by a different set of users or groups.

This element contains a `<search:accessControlEntries>` element.

<search:accessControlEntries>

Contains one or more `<search:accessControlEntry>` elements.

<search:accessControlEntry>

Provides a list of users and groups that have access to the source or are restricted from access. It contains these child elements:

```

<search:name>

```

<search:privilege>

<search:name>

Contains the name of a user or group that is valid for the currently active identity plug-in.

<search:privilege>

Set to GRANTED to allow access to the source, or set to DENIED to restrict access.

<search:authorizationPlugin>

Describes an authorization plug-in. See ["XML Description: User-Defined Sources"](#) on page 2-77.

<search:boundaryRules>

Contains one or more <search:boundaryRule> elements, each describing a boundary rule.

<search:boundaryRule>

Describes a boundary rule. It contains these child elements:

<search:ruleType>
<search:ruleOperation>
<search:rulePattern>

<search:ruleType>

Type of URL boundary rule. Set to one of these keywords:

- INCLUSION: The URL matches <search:rulePattern>.
- EXCLUSION: The URL does not match <search:rulePattern>.

<search:ruleOperation>

Matching operation for a search rule pattern. Set to one of these operations:

- CONTAINS: The URL contains the rule pattern for a case-insensitive match.
- STARTSWITH: The URL starts with the rule pattern for a case-insensitive match.
- ENDSWITH: The URL ends with the rule pattern for a case-insensitive match.
- REGEX: The URL contains the regular expression in a case-sensitive match.

<search:rulePattern>

The pattern of characters in the URL. You can use these special characters:

- Caret (^) denotes the beginning of a URL.
- Dollar sign (\$) denotes the end of a URL.
- A period (.) matches any one character.
- Question mark (?) before a character matches 0 or 1 occurrences of that character.
- Asterisk (*) before a pattern matches 0 or more occurrences of that pattern. Enclose the pattern in parentheses (), brackets [], or braces {}.
- A backslash (\) precedes a literal use of a special character, such as \? to match a question mark in a URL.

<search:metatagMappings>

Contains one or more <search:metatagMappings> elements.

<search:metatagMapping>

Contains a mapped pair of attributes in these child elements:

```
<search:documentAttr>
<search:searchAttr>
```

<search:documentAttr>

Identifies a document attribute by its name and data type. Document attributes are among the properties of a document.

Attribute	Value
name	Name of a document attribute. (Required)
type	Data type of the attribute: DATE, NUMBER, or STRING.

<search:searchAttr>

Identifies a search attribute by its name and data type. Search attributes are displayed to users in the Oracle SES Search interface.

Attribute	Value
name	Name of a search attribute. (Required)
type	Data type of the attribute: DATE, NUMBER, or STRING.

<search:crawlerSettings>

Configures the crawler. It contains these child elements:

```
<search:numThreads>
<search:languageDetection>
<search:defaultLanguage>
<search:crawlDepth>
<search:crawlTimeout>
<search:maxDocumentSize>
<search:preserveDocumentCache>
<search:defaultCharSet>
<search:servicePipeline>
<search:honorRobotsExclusion>
<search:indexDynamicPages>
<search:urlRewriter>
<search:httpCharSetOverride>
<search:cookies>
```

<search:numThreads>

Number of processes to use for crawling the source.

<search:languageDetection>

Controls the use of a language detector when the metadata for a document does not identify the language.

Attribute	value
enabled	Controls use of language detection when a source document does not indicate the language in the header. Set to <code>true</code> to enable language detection, or set to <code>false</code> otherwise. (Required)

<search:defaultLanguage>

Default language used by the crawler when the document language cannot be detected.

<search:crawlDepth>

Controls use of a limit on crawling nested links. It contains a `<search:limit>` element.

Attribute	Value
<code>hasLimit</code>	Controls whether the search limit is enforced. Set to <code>true</code> to impose the limit, or set to <code>false</code> otherwise. (Required)

<search:limit>

Contains the maximum number of nested links to be crawled.

<search:crawlTimeout>

Number of milliseconds for search results to be returned.

<search:maxDocumentSize>

Maximum document size in megabytes. Larger documents are not crawled.

<search:preserveDocumentCache>

Controls retention of the document cache after indexing.

Attribute	Value
<code>enabled</code>	Set to <code>true</code> to retain the cache, or set to <code>false</code> otherwise. (Required)

<search:defaultCharSet>

Code for the default character set, which is used when a source document does not identify its character set in the header. See [Table 2-4, "Crawlable Character Sets"](#).

<search:servicePipeline>

Controls use of a document service pipeline.

Attribute	Value
<code>enabled</code>	Set to <code>true</code> to use the pipeline, or set to <code>false</code> otherwise. When <code>true</code> , <code><search:servicePipeline></code> contains a <code><search:pipelineName></code> element.

<search:pipelineName>

Contains the name of a pipeline.

<search:honorRobotsExclusion>

Controls visits by robots to the Web site.

Attribute	Value
<code>enabled</code>	Set to <code>true</code> to exclude robots, or set to <code>false</code> otherwise.

<search:indexDynamicPages>

Controls whether dynamic pages are crawled and indexed.

Attribute	Value
<code>enabled</code>	Set to <code>true</code> to crawl dynamic pages, or set to <code>false</code> otherwise.

<search:urlRewriter>

Controls whether the URL Rewriter is used to filter and rewrite URL links. It contains these elements:

```
<search:urlRewriterClass>
<search:urlRewriterJar>
```

Attribute	Value
enabled	Set to <code>true</code> to use the URL Rewriter, or set to <code>false</code> otherwise.

<search:urlRewriterClass>

Contains the class name of the URL Rewriter.

<search:urlRewriterJar>

Contains the path to the JAR file for the URL Rewriter.

<search:httpCharSetOverride>

Controls the character set used for a Web page.

Attribute	Value
enabled	Set to <code>true</code> to exclude robots, or set to <code>false</code> otherwise.

<search:cookies>

Controls whether cookies are used to remember context. It contains these child elements:

```
<search:cookieContentInLog>
<search:maxCookieSize>
<search:maxCookies>
<search:maxCookiesPerHost>
```

Attribute	Value
enabled	Set to <code>true</code> to enable cookies (default), or <code>false</code> otherwise.

<search:cookieContentInLog>

Controls whether information about cookies appears in the log file.

Attribute	Value
enabled	Set to <code>true</code> to log cookie messages, or set to <code>false</code> otherwise (default).

<search:maxCookieSize>

Contains the maximum size in bytes of a cookie.

<search:maxCookies>

Contains the total number of cookies allowed in a crawl.

<search:maxCookiesPerHost>

Contains the maximum number of cookies permitted for a Web site.

<search:documentTypes>

Identifies the types of documents to be crawled. It contains one or more `<search:documentType>` elements.

<search:documentType>

Contains one or more `<search:mimeType>` elements.

<search:mimeType>

Contains the Internet media type of the content in the form *type/subtype*. See [Table 2-1, "Document Formats"](#).

<search:httpAuthentications>

Contains one or more `<search:httpAuthentication>` elements.

<search:httpAuthentication>

Describes HTTP authentication. For proxy authentication, it contains these elements:

```
<search:host>
<search:realm>
<search:username>
<search:password>
```

<search:host>

Contains the address of the target computer.

<search:realm>

Contains a name associated with the protected area of a Web site.

<search:username>

Contains the name of the log-in user.

<search:password>

Contains the password associated with the user name.

Attribute	Value
encrypted	Indicates whether the value of <code><search:password></code> is encrypted. Set to <code>true</code> if the password is encrypted, or set to <code>false</code> if it is plain text.

<search:htmlForms>

Contains one or more `<search:htmlForm>` elements, each one describing an HTML form.

<search:htmlForm>

Describes an HTML form. It contains these elements:

```
<search:name>
<search:formUrl>
<search:action>
<search:successUrl>
<search:formControls>
```

<search:name>

Contains the name of the HTML form object.

<search:formUrl>

Contains the Web address of the HTML form.

<search:action>

Contains the address where the browser sends the form.

<search:successUrl>

Contains the URL displayed after the user successfully submits the form.

<search:formControls>

Contains one or more `<search:formControl>` elements.

<search:formControl>

Describes a form control. It contains these elements:

```
<search:name>
<search:value>
<search:isPasswordField>
```

<search:name>

Contains the name of the form control.

<search:value>

Contains the value of the form control.

Attribute	Value
<code>encrypted</code>	Indicates whether the value of <code><search:value></code> is encrypted. Set to <code>true</code> if the value is encrypted, or set to <code>false</code> if it is plain text.

<search:isPasswordField>

Identifies whether the field contains a password. Set to `true` for a password field, or `false` otherwise.

<search:ssoAuthentication>

Describes OracleAS Single Sign-On authentication. It contains these elements:

```
<search:username>
<search:password>
```

Attribute	Value
<code>enabled</code>	Controls use of OracleAS Single Sign-On for authentication. Set to <code>true</code> to enable Single Sign-On, or <code>false</code> otherwise.

<search:username>

Contains a user name for OracleAS Single Sign-On.

<search:password>

Contains the password for the OracleAS Single Sign-On user.

Attribute	Value
<code>encrypted</code>	Indicates whether the value of <code><search:password></code> is encrypted. Set to <code>true</code> if the password is encrypted, or set to <code>false</code> if it is plain text.

Example 2-4 Web Source Description

This XML document describes a Web source.

```

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sources>
    <search:webSource>
      <search:name>this_websource</search:name>
      <search:startingUrls>
        <search:startingUrl>
          <search:url>http://www.example.com/</search:url>
        </search:startingUrl>
      </search:startingUrls>
      <search:aclPolicy>
        <search:noACL/>
      </search:aclPolicy>
      <search:boundaryRules>
        <search:boundaryRule>
          <search:ruleType>EXCLUSION</search:ruleType>
          <search:ruleOperation>STARTSWITH</search:ruleOperation>
          <search:rulePattern>
            <![CDATA[http://www.example.com?test=test val3]]>
          </search:rulePattern>
        </search:boundaryRule>
        <search:boundaryRule>
          <search:ruleType>INCLUSION</search:ruleType>
          <search:ruleOperation>CONTAINS</search:ruleOperation>
          <search:rulePattern>
            <![CDATA[http://www.example.com?test=test val]]>
          </search:rulePattern>
        </search:boundaryRule>
        <search:boundaryRule>
          <search:ruleType>INCLUSION</search:ruleType>
          <search:ruleOperation>REGEX</search:ruleOperation>
          <search:rulePattern>
            <![CDATA[^https?:/www\.example\.com(?:\:\d{1,5})?(?:$|/)]>
          </search:rulePattern>
        </search:boundaryRule>
      </search:boundaryRules>
      <search:metatagMappings>
        <search:metatagMapping>
          <search:documentAttr name="AUTHOR" type="STRING"/>
          <search:searchAttr name="Author" type="STRING"/>
        </search:metatagMapping>
        <search:metatagMapping>
          <search:documentAttr name="CREATOR" type="STRING"/>
          <search:searchAttr name="Author" type="STRING"/>
        </search:metatagMapping>
        <search:metatagMapping>
          <search:documentAttr name="DESCRIPTION" type="STRING"/>
          <search:searchAttr name="Description" type="STRING"/>
        </search:metatagMapping>
        <search:metatagMapping>
          <search:documentAttr name="KEYWORD" type="STRING"/>
          <search:searchAttr name="Keywords" type="STRING"/>
        </search:metatagMapping>
        <search:metatagMapping>
          <search:documentAttr name="KEYWORDS" type="STRING"/>
          <search:searchAttr name="Keywords" type="STRING"/>
        </search:metatagMapping>
        <search:metatagMapping>
          <search:documentAttr name="SUBJECT" type="STRING"/>
        </search:metatagMapping>
      </search:metatagMappings>
    </search:webSource>
  </search:sources>
</search:config>

```



```

        <search:searchAttr name="Subject" type="STRING"/>
    </search:metatagMapping>
    <search:metatagMapping>
        <search:documentAttr name="SUBJECTS" type="STRING"/>
        <search:searchAttr name="Subject" type="STRING"/>
    </search:metatagMapping>
</search:metatagMappings>
<search:crawlerSettings>
    <search:numThreads>7</search:numThreads>
    <search:languageDetection enabled="true"/>
    <search:defaultLanguage>fr</search:defaultLanguage>
    <search:crawlDepth haslimit="true">
        <search:limit>2</search:limit>
    </search:crawlDepth>
    <search:crawlTimeout>100</search:crawlTimeout>
    <search:maxDocumentSize>1000</search:maxDocumentSize>
    <search:preserveDocumentCache enabled="true"/>
    <search:defaultCharSet>JIS</search:defaultCharSet>
    <search:servicePipeline enabled="false"/>
    <search:honorRobotsExclusion enabled="false"/>
    <search:indexDynamicPages enabled="true"/>
    <search:urlRewriter enabled="false"/>
    <search:httpCharSetOverride enabled="false"/>
    <search:cookies enabled="true">
        <search:cookieContentInLog enabled="false"/>
        <search:maxCookieSize>1</search:maxCookieSize>
        <search:maxCookies>2</search:maxCookies>
        <search:maxCookiesPerHost>3</search:maxCookiesPerHost>
    </search:cookies>
</search:crawlerSettings>
<search:documentTypes>
    <search:documentType>
        <search:mimeType>application/msword</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>application/pdf</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>application/x-msexcel</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>application/x-mspowerpoint</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>text/html</search:mimeType>
    </search:documentType>
    <search:documentType>
        <search:mimeType>text/plain</search:mimeType>
    </search:documentType>
</search:documentTypes>
<search:httpAuthentications>
    <search:httpAuthentication>
        <search:host>testhost1</search:host>
        <search:realm>testrealm1</search:realm>
        <search:username>testusername1</search:username>
        <search:password encrypted="false">
            password
        </search:password>
    </search:httpAuthentication>
</search:httpAuthentications>

```

```
<search:htmlForms>
  <search:htmlForm>
    <search:name>testformname1</search:name>
    <search:formUrl>http://test2.oracle.com</search:formUrl>
    <search:action>test</search:action>
    <search:successUrl>
      http://successurl.oracle.com
    </search:successUrl>
    <search:formControls>
      <search:formControl>
        <search:name>testcontrol1</search:name>
        <search:value encrypted="false">testvalue1</search:value>
        <search:isPasswordField>false</search:isPasswordField>
      </search:formControl>
      <search:formControl>
        <search:name>testcontrol2</search:name>
        <search:value encrypted="false">
          this_value
        </search:value>
        <search:isPasswordField>true</search:isPasswordField>
      </search:formControl>
    </search:formControls>
  </search:htmlForm>
</search:htmlForms>
<search:ssoAuthentication enabled="true">
  <search:username>testssso</search:username>
  <search:password encrypted="false">
    password
  </search:password>
</search:ssoAuthentication>
</search:webSource>
</search:sources>
</search:config>
```

sourceGroup

A source group consists of one or more sources. When entering a search, users can select the source groups to search instead of searching all available sources. A source can belong to multiple source groups.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Page

Global Settings - Translate Source Group Name
Search - Source Groups - Create or Edit Source Group

XML Description

The <search:sourceGroups> element describes source groups:

```
<search:sourceGroups>
  <search:sourceGroup>
    <search:name>
    <search:translations>
    <search:assignedSources>
      <search:assignedSource>
```

Schema Descriptions

<search:sourceGroups>

Contains one or more <search:sourceGroup> elements, each defining a source group.

<search:sourceGroup>

Describes a source group. It contains these elements:

```
<search:name>
<search:translations>
<search:assignedSources>
```

<search:name>

Contains the name of the source group. (Required)

<search:translations>

Contains translations of the object name for display. See ["Providing Translations of Object Names"](#) on page 2-6.

<search:assignedSources>

Contains one or more <search:assignedSource> elements, each identifying a source assigned to this source group.

<search:assignedSource>

Contains the name of a source in this source group.

Example

This XML document defines two source groups, Web and Calendar:

```
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sourceGroups>
    <search:sourceGroup>
      <search:name>Web</search:name>
    </search:sourceGroup>
    <search:sourceGroup>
      <search:name>Calendar</search:name>
      <search:translations>
        <search:translation language="de">
          <search:translatedValue>Kalender</search:translatedValue>
        </search:translation>
        <search:translation language="fi">
          <search:translatedValue>kalenteri</search:translatedValue>
        </search:translation>
        <search:translation language="es">
          <search:translatedValue>calendario</search:translatedValue>
        </search:translation>
        <search:translation language="pt-br">
          <search:translatedValue>calendario</search:translatedValue>
        </search:translation>
      </search:translations>
    </search:sourceGroup>
  </search:sourceGroups>
</search:config>
```

sourceType

A source type identifies where the information for a source is stored, such as on a Web site or in a database table. Oracle SES provides several built-in source types.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Page

Global Settings - Source Types - Create or Update Source Type

XML Description

The `<search:sourceTypes>` element describes the source types:

```
<search:sourceTypes>
  <search:sourceType>
    <search:name>
    <search:managerClassName>
    <search:jarFilePath>
    <search:description>
    <search:securityCapability>
    <search:parameterInfos>
      <search:parameterInfo>
        <search:defaultValue>
        <search:encrypted>
        <search:description>
```

Element Descriptions

<search:sourceTypes>

Describes all source types. It contains one or more <search:sourceType> elements, each defining a source type.

<search:sourceType>

Describes a source type. It contains these elements:

```
<search:name>
<search:managerClassName>
<search:jarFilePath>
<search:description>
<search:securityCapability>
<search:parameterInfos>
```

<search:name>

Contains the name of the source type.

<search:managerClassName>

Contains the name of the plug-in manager Java class.

<search:jarFilePath>

Contains the qualified name of the jar file. Paths can be absolute or relative to the *ORACLE_HOME/search/lib/plugins* directory.

<search:description>

Contains a description of the source type.

<search:securityCapability>

Contains one of these values from the plug-in: *IDENTITY_BASED*, *USER_DEFINED*, or *UNKNOWN*. (Read only)

<search:parameterInfos>

Contains one or more <search:parameterInfo> elements, each describing a parameter of the source type.

<search:parameterInfo>

Describes a parameter. It contains these elements:

```
<search:defaultValue>
<search:encrypted>
<search:description>
```

Attribute	Value
Name	Name of the parameter. (Required)

<search:defaultValue>

Default value of the parameter.

<search:encrypted>

Indicates whether the parameter represents a value that should be encrypted. Set to *true* to encrypt the value, or set to *false* otherwise. The default value is *false*. (Optional)

Example

This XML document describes the Oracle Content Database source type:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:sourceTypes>
    <search:sourceType>
      <search:name>Oracle Content Database</search:name>
      <search:managerClassName>
        oracle.search.plugin.ocs.cservices.OCSCSPluginMgr
      </search:managerClassName>
      <search:jarFilePath>cservices/ocscsrvV2.jar</search:jarFilePath>
      <search:description>
        Oracle Content Database crawler plug-in
      </search:description>
      <search:securityCapability>USER_DEFINED</search:securityCapability>
      <search:parameterInfos>
        <search:parameterInfo name="CDB Server public key alias">
          <search:encrypted>>false</search:encrypted>
          <search:description>
            Oracle Content Database Server public key alias
          </search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Crawl only">
          <search:defaultValue>>false</search:defaultValue>
          <search:encrypted>>false</search:encrypted>
          <search:description>
            True will perform a crawl without indexing the documents
          </search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Depth">
          <search:defaultValue>-1</search:defaultValue>
          <search:encrypted>>false</search:encrypted>
          <search:description>
            Depth from starting paths ("-1" for no limit)
          </search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Entity name">
          <search:encrypted>>false</search:encrypted>
          <search:description>
            Name of the trusted entity in Oracle Internet Directory (OID)
          </search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Entity password">
          <search:encrypted>>true</search:encrypted>
          <search:description>
            Password of the trusted entity in OID
          </search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Oracle Content Database URL">
          <search:encrypted>>false</search:encrypted>
          <search:description>Oracle Content Database Web services endpoint;
for example, "http://contentserver:7777/content"</search:description>
        </search:parameterInfo>
        <search:parameterInfo name="Oracle Content Database Version">
          <search:defaultValue>10.1.2.3.0</search:defaultValue>
          <search:encrypted>>false</search:encrypted>
          <search:description>Oracle Content Database version; for example,
"10.1.2.3.0"</search:description>
      </search:parameterInfos>
    </search:sourceType>
  </search:sourceTypes>
</search:config>
```

```

</search:parameterInfo>
<search:parameterInfo name="Oracle Content Database admin user">
  <search:encrypted>>false</search:encrypted>
  <search:description>Name of administrator user for Oracle Content
Database; for example, orcladmin</search:description>
</search:parameterInfo>
<search:parameterInfo name="SES keystore location">
  <search:encrypted>>false</search:encrypted>
  <search:description>
    SES keystore location for WS security
  </search:description>
</search:parameterInfo>
<search:parameterInfo name="SES keystore password">
  <search:encrypted>>true</search:encrypted>
  <search:description>SES keystore password</search:description>
</search:parameterInfo>
<search:parameterInfo name="SES keystore type">
  <search:encrypted>>false</search:encrypted>
  <search:description>SES keystore type</search:description>
</search:parameterInfo>
<search:parameterInfo name="SES private key alias">
  <search:encrypted>>false</search:encrypted>
  <search:description>
    SES client private key alias
  </search:description>
</search:parameterInfo>
<search:parameterInfo name="SES private key password">
  <search:encrypted>>true</search:encrypted>
  <search:description>
    SES client private key password
  </search:description>
</search:parameterInfo>
<search:parameterInfo name="Starting paths">
  <search:defaultValue></search:defaultValue>
  <search:encrypted>>false</search:encrypted>
  <search:description>
    Paths (not encoded) to start crawling (separated by ";" )
  </search:description>
</search:parameterInfo>
<search:parameterInfo name="Use e-mail for authorization">
  <search:defaultValue>>false</search:defaultValue>
  <search:encrypted>>false</search:encrypted>
  <search:description>Use e-mail to resolve the user privilege. Set
this to true if the Oracle Internet Directory has been configured to use "mail" as
the nickname attribute.</search:description>
</search:parameterInfo>
</search:parameterInfos>
</search:sourceType>
</search:sourceTypes>
</search:config>

```


spaceCalculator

The space calculator checks the disk usage of Oracle SES storage areas to ensure they do not exceed the current space quotas. When a storage area approaches its quota, the space calculator signals a warning. When a storage area exceeds its quota, the space calculator signals an alert. You can configure how frequently the space calculator runs, or you can start it manually.

See Also: ["Disk Space Management: Quotas and Alerts"](#) on page 2-12

Object Type

Universal

State Properties

Property	Value
endTime	Date and time the space calculator completed its assessment of disk usage
error	Error numbers and messages when the space calculator did not run successfully
startTime	Date and time the space calculator last began its assessment of disk usage
status	EXECUTING, FAILED, PARTIALLY_FAILED, SCHEDULED, or SUCCESS

Supported Operations

activate
deactivate
export
getState
start
update

Administration GUI Page

None

XML Description

The `<search:spaceCalculator>` element configures the space calculator:

```
<search:spaceCalculator>
  <search:frequency>
<!-- For hourly calculations -->
    <search:hourly>
      <search:hoursBtwnLaunches>
<!-- For daily calculations -->
    <search:daily>
      <search:daysBtwnLaunches>
      <search:startHour>
<!-- For weekly calculations -->
    <search:weekly>
      <search:weeksBtwnLaunches>
      <search:startDayOfWeek>
```

```
    <search:startHour>
<!-- For monthly calculations -->
    <search:monthly>
        <search:monthsBtwLaunches>
        <search:startDayOfMonth>
        <search:startHour>
```

Element Descriptions

<search:spaceCalculator>

Contains a `<search:frequency>` element.

<search:frequency>

Describes the space calculator schedule. It contains one of these elements:

```
<search:hourly>
<search:daily>
<search:weekly>
<search:monthly>
```

<search:hourly>

Describes an hourly schedule. It contains a `<search:hoursBtwLaunches>` element.

<search:hoursBtwLaunches>

The number of hours between optimizations.

<search:daily>

Describes a daily schedule. It contains these elements:

```
<search:daysBtwLaunches>
<search:startHour>
```

<search:daysBtwLaunches>

The number of days between space calculations.

<search:startHour>

The time the crawl begins using a 24-hour clock, such as 9 for 9:00 a.m. or 23 for 11:00 p.m.

<search:weekly>

Describes a weekly schedule. It contains these elements:

```
<search:weeksBtwLaunches>
<search:startDayOfWeek>
<search:startHour>
```

<search:weeksBtwLaunches>

The number of weeks between space calculations.

<search:startDayOfWeek>

The day of the week that the crawl begins, such as MONDAY or TUESDAY.

<search:monthly>

Describes a monthly schedule. It contains these elements:

```
<search:monthsBtwLaunches>
<search:startDayOfMonth>
<search:startHour>
```

<search:monthsBtwnLaunches>

The number of time periods between starting a calculation.

<search:startDayOfMonth>

An integer value for the day of the month that the calculation begins, such as 1 or 15.

Example

This example updates the space calculator to a weekly schedule:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:spaceCalculator>
    <search:frequency>
      <search:weekly>
        <search:weeksBtwnLaunches>1</search:weeksBtwnLaunches>
        <search:startDayOfWeek>THURSDAY</search:startDayOfWeek>
        <search:startHour>3</search:startHour>
      </search:weekly>
    </search:frequency>
  </search:spaceCalculator>
</search:config>
```

storageArea

A storage area is a physical structure, such as a file, that Oracle SES uses to store data and metadata. You can set the maximum size of a storage area. Otherwise, it grows as required by your data sources and may consume all available resources.

See Also:

- ["Partitioning for Parallel Query"](#) on page 2-10
- ["Disk Space Management: Quotas and Alerts"](#) on page 2-12

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll

Administration GUI Page

None

XML Description

A `<search:storageAreas>` element describes a storage area:

```
<search:storageAreas>
  <search:storageArea>
    <search:name>
    <search:description>
    <search:usage>
    <search:locations>
      <search:location>
        <search:path>
```

```

<search:device>
<search:preAllocatedSpace>
<search:quota>
<search:currentSize>

```

Element Contents:

<search:storageAreas>

Contains one or more <search:storageArea> elements, each defining a storage area for use by Oracle SES.

<search:storageArea>

Describes a storage area. It contains these elements:

```

<search:name>
<search:description>
<search:usage>
<search:locations>

```

<search:name>

Name of the storage area. (Required)

<search:description>

Description of the storage area. (Required)

<search:usage>

A usage type. You can create only the PARTITION storage type. (Required)

- PARTITION: Used by the crawler to store the document index.
- CACHE_FILE: Used to store the secure cache. You can update the quota, but no other parameters. You cannot create or delete the cache file storage area.
- SYSTEM: Internal storage area. You can update the quota, but no other parameters. You cannot create or delete the system storage area.

<search:locations>

Contains a <search:location> element.

<search:location>

Describes the physical attributes of the storage area. (Required)

It contains these elements:

```

<search:path>
<search:device>
<search:preAllocatedSpace>
<search:quota>
<search:currentSize>

```

<search:path>

Fully qualified path for the storage area. The full path must exist.

<search:device>

Contains DEFAULT or the name of a hard drive, CD-ROM, network share, or similar storage hardware. If you omit this element or enter DEFAULT, then Oracle SES creates a tablespace in the default tablespace directory for the Oracle Database instance.

<search:preAllocatedSpace>

The initial size of the file in megabytes.

<search:quota>

The maximum size of the storage area in megabytes. (Required)

To remove a quota that was set previously, enter -1. Oracle SES does not check the size of a storage area after its quota is removed.

<search:currentSize>

The current size of the file in megabytes. (Read-only)

Attribute	Value
lastRefreshDate	Date of the most recent refresh, when the current size was calculated.

Example

This XML document creates a tablespace named PROSPERITY in a file named /ses_install/data/PROSPERITY_1.dbf for storing the Secure Cache. It has an initial size of 4MB and a maximum size of 64MB.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:storageAreas>
    <search:storageArea>
      <search:name>Prosperity</search:name>
      <search:description>Secure cache new storage</search:description>
      <search:usage>CACHE_FILE</search:usage>
      <search:locations>
        <search:location>
          <search:path>/ses_install/data/</search:path>
          <search:preAllocatedSpace>4</search:preAllocatedSpace>
          <search:quota>64</search:quota>
        </search:location>
      </search:locations>
    </search:storageArea>
  </search:storageAreas>
</search:config>
```

suggLink

Suggested links direct users to a designated Web site for particular query keywords. For example, a suggested link might be `http://www.oracle.com/technology/documentation/ses.html` for 'Oracle Secure Enterprise Search documentation', 'Enterprise Search documentation', and 'Search documentation'.

Object Type

Creatable

Object Key

keyword linkUrl

Object Key Command Syntax

```
--KEYWORD=keyword --LINK_URL=url
```

State Properties

None

Supported Operations

```
create
createAll
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
update
updateAll
```

Administration GUI Page

Search - Suggested Links

XML Description

The `<search:suggLinks>` element describes suggested links:

```
<search:suggLinks>
  <search:suggLink>
    <search:keyword>
    <search:linkUrl>
    <search:linkText>
```

Element Descriptions

<search:suggLinks>

Contains one or more `<search:suggLink>` elements, each describing a suggested link.

<search:suggLink>

Describes a suggested link. It contains one of each of these child elements:

```
<search:keyword>
<search:linkUrl>
<search:linkText>
```

<search:keyword>

A word or phrase with optional operators that identifies which search queries display this suggested link. (Required)

Do not enter special characters, such as #, \$, =, &. You can include the following operators:

Operation	Syntax	Example
about	ABOUT (<i>term</i>)	about(dogs)
and	<i>term</i> AND <i>term</i>	dog and cat
near	<i>term</i> ; <i>term</i>	dog ; cat
or	<i>term</i> OR <i>term</i>	dog or cat
phrase	<i>phrase</i>	dog sled
stem	<i>\$term</i>	\$dog
thesaurus	{BT NT SYN} (<i>term</i>)	SYN(dog)
within	<i>term</i> WITHIN <i>term</i>	dog within title

<search:linkUrl>

A link to the suggested page, which appears in the result list., such as <http://www.example.com>. (Required)

<search:linkText>

The linked text that appears in the result list, such as Example Corp. (Required)

Example

This XML document defines a suggested link for a query on the term "oracle":

```
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:suggLinks>
    <search:suggLink>
      <search:keyword>oracle</search:keyword>
      <search:linkUrl>http://www.oracle.com</search:linkUrl>
      <search:linkText>Oracle</search:linkText>
    </search:suggLink>
  </search:suggLinks>
</search:config>
```

task

A task controls Oracle SES space-consuming background processes. It can disable, stop, or resume these processes as part of the space management system that prevents Oracle SES from exceeding its disk quotas.

Object Type

Creatable by Oracle SES space management

Object Key

name

Object Key Command Syntax

```
--NAME={ stopAllSpaceConsumingTasks | resumeAllSpaceConsumingTasks }
```

```
-n { stopAllSpaceConsumingTasks | resumeAllSpaceConsumingTasks }
```

State Properties

The following table describes state properties for `stopAllSpaceConsumingTasks`:

Property	Value
<code>disabledSchedules</code>	Any disabled crawler schedules
<code>endTime</code>	End date and time of the task
<code>error</code>	Any error messages
<code>startTime</code>	Start date and time of the task
<code>status</code>	DISABLED, EXECUTING, FAILED, LAUNCHING, PARTIALLY FAILED, SCHEDULED, or SUCCESS
<code>stoppedSchedules</code>	Any stopped crawler schedules

Supported Operations

`getState`
`start`

Administration GUI Page

None

thesaurus

A thesaurus is a list of terms or phrases with relationships specified among them, such as a synonym, a broader term, and a narrower term. When a user issues a search query, Oracle SES can expand the search results to include matches for the related terms.

A thesaurus contains domain-specific knowledge. You can build a thesaurus, buy an industrial-specific thesaurus, or use utilities to extract a thesaurus from a specific corpus of documents. The thesaurus must be compliant with both the ISO-2788 and ANSI Z39.19(1993) standards.

A thesaurus must be loaded in Oracle SES for thesaurus-based query expansion. If no thesaurus is loaded or if the specified term or phrase cannot be found in the loaded thesaurus, then query expansion is not possible. Oracle SES only returns documents containing the original term or phrase. The default expansion level is one.

The proper encoding of an XML document for thesaurus configuration is UTF-8, which is the Oracle SES default language setting. Ensure that the NLS_LANG environment variable setting is consistent with the XML document encoding.

Object Type

Creatable

Object Key

name

Object Key Command Syntax

--NAME=*object_name*

-n *object_name*

State Properties

None

Supported Operations

create
delete
export
getAllObjectKeys
update

Administration GUI Page

None

XML Description

The `<search:thesauruses>` element defines a thesaurus:

```
<search:thesauruses>  
  <search:thesaurus>  
    <search:name>  
    <search:thesaurusContent>
```

Element Descriptions

<search:thesauruses>

Contains a <search:thesaurus> element, which describes a thesaurus.

<search:thesaurus>

Describes a thesaurus. It contains these child elements:

```
<search:name>
<search:thesaurusContent>
```

<search:name>

The name of the thesaurus. This name must be DEFAULT. (Required)

<search:thesaurusContent>

The thesaurus content. (Required)

Enter each term on a separate line within a CDATA element. You can identify broader terms (BT), narrower terms (NT) and synonyms (SYN). Note the one-space indentation of the related terms:

```
dog
  BT mammal
  NT domestic dog
  NT wild dog
  SYN canine
```

Example

This XML document defines the default thesaurus:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:thesauruses>
    <search:thesaurus>
      <search:name>DEFAULT</search:name>
      <search:thesaurusContent>
<![CDATA[
cat
  SYN feline
  NT domestic cat
  NT wild cat
  BT mammal
mammal
  BT animal
domestic cat
  NT Persian cat
  NT Siamese cat
wild cat
  NT tiger
tiger
  NT Bengal tiger
dog
  BT mammal
  NT domestic dog
  NT wild dog
  SYN canine
domestic dog
  NT German Shepard
wild dog
```

```
NT Dingo
]]>
  </search:thesaurusContent>
    </search:thesaurus>
  </search:thesauruses>
</search:config>
```

searchadmin Commands

This chapter describes the `searchadmin` commands. It contains these topics:

- [Alphabetic List of searchadmin Commands](#)
- [Entering Special Characters](#)

Alphabetic List of searchadmin Commands

ACDEGSU

A

```
activate clustering
activate clusterTree
activate identityPlugin
activate indexOptimizer
activate resultList
activate schedule
activate skinBundle
activate spaceCalculator
```

C

```
create altWord
create clusterTree
create identityPlugin
create proxyLogin
create schedule
create searchAttr
create skinBundle
create source
create sourceGroup
create sourceType
create storageArea
create suggLink
create thesaurus
createAll altWord
createAll clusterTree
createAll identityPlugin
createAll proxyLogin
createAll schedule
createAll searchAttr
createAll source
createAll sourceGroup
createAll sourceType
createAll storageArea
```

createAll suggLink

D

deactivate clustering
deactivate clusterTree
deactivate identityPlugin
deactivate indexOptimizer
deactivate resultList
deactivate schedule
deactivate skinBundle
deactivate spaceCalculator
delete creatable_type
deleteAll creatable_type
deleteList creatable_type

E

export creatable_type
export universal_type
exportAll creatable_type
exportList creatable_type

G

getAllObjectKeys
getAPIVersion
getAllStates clusterTree
getAllStates identityPlugin
getAllStates schedule
getAllStates skinBundle
getState clustering
getState clusterTree
getState identityPlugin
getState index
getState indexOptimizer
getState resultList
getState schedule
getState skinBundle
getState spaceCalculator
getState task
getStateList clusterTree
getStateList identityPlugin
getStateList schedule
getStateList skinBundle

S

start indexOptimizer
start schedule
start task
stop schedule
start spaceCalculator

U

update alert
update altWord
update clustering
update clusterTree
update crawlerSettings
update index
update indexOptimizer

```
update partitionConfig
update proxyLogin
update queryConfig
update resultList
update schedule
update searchAttr
update skinBundle
update source
update sourceGroup
update sourceType
update spaceCalculator
update storageArea
update suggLink
update thesaurus
updateAll alert
updateAll altWord
updateAll clusterTree
updateAll proxyLogin
updateAll schedule
updateAll searchAttr
updateAll source
updateAll sourceGroup
updateAll sourceType
updateAll suggLink
```

Entering Special Characters

To enter special characters as part of a command, you may have to enclose the value in quotes, or precede the character with backslashes as an escape, or both.

- Enclose values containing spaces in quotes, such as `--NAME="Special Collection"`.
- To use the percent (%) and underscore (_) wildcard characters as literals for key patterns in operations such as `createAll` and `deleteAll`, escape the character with two backslashes, such as `--NAME=web_source` for an object named `web_source`. In single job mode, also enclose the values containing these characters in quotes, such as `--NAME="web_source"`. Otherwise, the underscore matches any single character.
- When using the short form for arguments, escape literal values starting with a dash with two backslashes in single job mode, such as `-n \\-mysource`. In session mode, enclose the values in quotes or escape the dash with one backslash. For example, `-n \-mysource` or `-n "-mysource"`. Alternatively, use the long form for arguments, such as `--NAME=-mysource`.

See "[Command Interface](#)" on page 1-1 for a discussion of session mode and single job mode.

activate clustering

Activates clustering.

Syntax

```
activate clustering
```

Example

This example activates clustering:

```
SES>activate clustering
```

```
The object "clustering" was successfully activated.
```

activate clusterTree

Activates a cluster tree.

A `clusterTree` object is active when it is created. You must activate a `clusterTree` only after deactivating it.

Syntax

```
activate clusterTree --NAME=object_name
```

or

```
activate clusterTree -n object_name]
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

Example

This example activates a cluster tree named `Topic Tree`, which was previously created and deactivated:

```
SES>activate clusterTree --NAME="Topic Tree"
```

The object "[name=Topic Tree]" was successfully activated.

activate identityPlugin

Activates an identity plug-in.

Only one identity plug-in can be active at a time. To change identity plug-ins, deactivate the current one before activating a different identity plug-in. Otherwise, an error results. An identity plug-in is inactive when it is created.

Syntax

```
activate identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class [--INPUT_
FILE=xml_filename --ENCRYPT_KEY=key]
```

or

```
activate identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class [-i xml_
filename -e key]
```

Parameters

jar_filename

Content of a <search:jarFilePath> element in the XML document.

class

Content of a <search:managerClassName> element in the XML document.

xml_filename

Path to an XML document that contains parameter settings for the object. See [identityPlugin](#) on page 2-38.

key

Decryption key for passwords in *xml_filename*. If the plug-in description has been exported from Oracle SES, use the same key.

The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example activates an identity plug-in with the configuration settings in *identity.xml*:

```
SES>activate identityPlugin --JAR_FILE=OIDPlugins.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.oid.OIDPluginManager --INPUT_
FILE=identity.xml --ENCRYPT_KEY=key2decrypt
```

The object "[jarFilePath=OIDPlugins.jar, managerClassName=oracle.search.plugin.security.identity.oid.OIDPluginManager]" was successfully activated.

activate indexOptimizer

Activates the index optimizer schedule.

See Also

[start indexOptimizer](#)

Syntax

```
activate indexOptimizer
```

Example

This example activates the index optimizer:

```
SES>activate indexOptimizer
```

The object "indexOptimizer" was successfully activated.

activate resultList

Enables the global advanced result configuration option.

Syntax

```
activate resultList
```

Example

This example activates the result list:

```
SES>activate resultList
```

The object "resultList" was successfully activated.

activate schedule

Activates a schedule.

See Also

[start schedule](#)

Syntax

```
activate schedule --NAME=object_name [--INPUT_FILE=xml_filename]
```

or

```
activate schedule -n object_name [-i xml_filename]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that describes the object. See [schedule](#) on page 2-59.

Example

This example activates a schedule named Mailing List Schedule:

```
SES>activate schedule --NAME="Mailing List Schedule"
```

The object "[name=Mailing List Schedule]" was successfully activated.

activate skinBundle

Activates a skin bundle.

See Also

["Search Interface Customization: Skin Bundles"](#) on page 2-13

Syntax

```
activate skinBundle --NAME=object_name [--INPUT_FILE=xml_filename] [--ATTACHMENT_
LIST=list_filename]
```

or

```
activate skinBundle -n object_name [-i xml_filename] [-h list_filename]
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that describes the object. See [skinBundle](#) on page 2-66.

list_filename

Path to the text file that lists the files in the skin bundle. See the Notes for [create skinBundle](#) on page 3-18.

Example

This example activates a skin bundle named acme:

```
SES>activate skinBundle --NAME=acme
```

The object "[name=acme]" was successfully activated

activate spaceCalculator

Activates the space calculator schedule.

The space calculator is active by default. You can activate it only after deactivating it.

See Also

[start spaceCalculator](#)

Syntax

```
activate spaceCalculator
```

Example

This example activates the space calculator:

```
SES>activate spaceCalculator
```

```
The object "spaceCalculator" was successfully activated.
```

create altWord

Creates an alternate word pair from an XML description.

See Also

[createAll altWord](#)

Syntax

```
create altWord --KEYWORD=keyword --ALT_KEYWORD=alt_keyword --INPUT_FILE=xml_  
filename
```

or

```
create altWord --KEYWORD=keyword --ALT_KEYWORD=alt_keyword -i xml_filename
```

Parameters

keyword

Content of a `<search:keyword>` element in the XML document.

alt_keyword

Content of a `<search:altKeyword>` element in the XML document.

xml_filename

Path to the XML document that defines the object. See [altWord](#) on page 2-26.

Example

This example creates an alternate word for RAC. No other objects in the XML document are created.

```
SES>create altWord --KEYWORD=rac --ALT_KEYWORD="Real Application Clusters"  
--INPUT_FILE=altwords.xml
```

The object "[keyword=rac, altKeyword=Real Application Clusters]" was successfully created.

create clusterTree

Creates a cluster tree from an XML description.

See Also

[createAll clusterTree](#)

Syntax

```
create clusterTree --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create clusterTree -n object_name -i xml_filename
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [clusterTree](#) on page 2-30.

Example

This example creates a metadata cluster tree defined in clustertree.xml. No other objects in the document are created.

```
SES>create clusterTree --NAME="Metadata Tree" --INPUT_FILE=clustertree.xml
```

The object "[name=Metadata Tree]" was successfully created.

create identityPlugin

Creates an identity plug-in from an XML description.

See Also

[createAll identityPlugin](#)

Syntax

```
create identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class --INPUT_
FILE=xml_filename]
```

or

```
create identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class -i xml_
filename]
```

Parameters

jar_filename

Content of a <search:jarFilePath> element in the XML document.

class

Content of a <search:managerClassName> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [identityPlugin](#) on page 2-38.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example creates an identity plug-in defined in identity.xml.

```
SES>create identityPlugin --JAR_FILE=OIDPlugins.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.oid.OIDPluginManager --INPUT_
FILE=identity.xml
```

The object "[jarFilePath=OIDPlugins.jar, managerClassName=oracle.search.plugin.security.identity.oid.OIDPluginManager]" was successfully created.

create proxyLogin

Creates a proxy log-in (federation trusted entity) from an XML description.

See Also

[createAll proxyLogin](#)

Syntax

```
create proxyLogin --NAME=object_name --INPUT_FILE=xml_filename [--ENCRYPT_KEY=key]
or
create proxyLogin -n object_name -i xml_filename [-e key]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [proxyLogin](#) on page 2-49.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example creates `this_proxy` as defined in `proxy.xml`.

The user is prompted for a password after omitting the `--ENCRYPT_KEY` option.

```
SES>create proxyLogin --NAME=this_proxy --INPUT_FILE=proxy.xml
Enter encryption key:
```

The object "[name=this_proxy]" was successfully created.

create schedule

Creates a schedule from an XML description. You must create the source before creating the schedule. See [create source](#) on page 3-19.

See Also

[createAll schedule](#)

Syntax

```
create schedule --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create schedule -n object_name -i xml_filename
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that defines the object. See [schedule](#) on page 2-59.

Example

This example creates the Doc Library schedule as defined in schedule.xml:

```
SES>create schedule --NAME="Doc Library" --INPUT_FILE=schedule.xml
```

```
The object "[name=Doc Library]" was successfully created.
```

create searchAttr

Creates a custom search attribute from an XML description.

See Also

[createAll searchAttr](#)

Syntax

```
create searchAttr --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create searchAttr -n object_name -i xml_filename
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that defines the object. See [searchAttr](#) on page 2-63.

Example

This example creates the `Copyright` search attribute:

```
SES>create searchAttr --NAME=Copyright --INPUT_FILE=searchattrs.xml
```

The object "[name=Copyright]" was successfully created.

create skinBundle

Creates a skin bundle from an XML description and a structured directory of files.

See Also

["Search Interface Customization: Skin Bundles"](#) on page 2-13

Syntax

```
create skinBundle --NAME=object_name --INPUT_FILE=xml_filename --ATTACHMENT_
LIST=list_filename
```

or

```
create searchAttr -n object_name -i xml_filename -h list_filename
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that defines the object. See [skinBundle](#) on page 2-66.

list_filename

Path to the text file that lists the files in the skin bundle. See the Notes.

Notes

To identify the location of the files composing the skin bundle, create a text file with a line for each file in this format:

```
resource_name::local_file_path
```

Where:

resource_name identifies the location of the file in the skin bundle. Use this resource name in the XML description of the skin bundle.

local_file_path is the fully qualified name of the file.

For example:

```
templates/query.ftl::/scratch/skins/Holiday/templates/query.ftl
assets/images/logo.gif::/scratch/skins/Holiday/assets/images/logo.gif
```

Example

This example creates the `acme` skin bundle from the description in `skins.xml` and the files identified in `skins/resources.lst`:

```
SES>create skinBundle --NAME=acme --INPUT_FILE=skins.xml --ATTACHMENT_
LIST=skins/resources.lst
```

The object "[name=acme]" was successfully created.

create source

Creates a source from an XML description.

See Also

[createAll source](#)

Syntax

```
create source --NAME=object_name --INPUT_FILE=xml_filename [--ENCRYPT_KEY=key]
```

or

```
create source -n object_name -i xml_filename [-e key]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [source](#) on page 2-68.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example creates the Doc Library source defined in sources.xml:

```
SES>create source --NAME="Doc Library" --INPUT_FILE=sources.xml --ENCRYPT_
KEY=key2encrypt
```

The object "[name=Doc Library]" was successfully created.

create sourceGroup

Creates a source group from an XML description.

See Also

[createAll sourceGroup](#)

Syntax

```
create sourceGroup --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create sourceGroup -n object_name -i xml_filename
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [sourceGroup](#) on page 2-93.

Example

This example creates the Mail source group defined in sourcegroups.xml:

```
SES>create sourceGroup --NAME=Mail --INPUT_FILE=sourcegroups.xml
```

The object "[name=Mail]" was successfully created.

create sourceType

Creates a source type from an XML description.

See Also

[createAll sourceType](#)

Syntax

```
create sourceType --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create sourceType -n object_name -i xml_filename
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that defines the object. See [sourceType](#) on page 2-95.

Example

This example creates the `New Agent` source type defined in `sourcetype.xml`.

```
SES>create sourceType --NAME="New Agent" --INPUT_FILE=sourcetype.xml
```

The object "[name=New Agent]" was successfully created.

create storageArea

Creates a storage area from an XML description.

See Also

[createAll storageArea](#)

Syntax

```
create storageArea --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create storageArea -n object_name -i xml_filename
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [storageArea](#) on page 2-102.

Example

This example creates a storage area named Prosperity:

```
SES>create storageArea --NAME=Prosperity --INPUT_FILE=storage.xml
```

The object "[name=Prosperity]" was successfully created.

create suggLink

Creates a suggested link from an XML description.

See Also

[createAll suggLink](#)

Syntax

```
create suggLink --KEYWORD=keyword --LINK_URL=url --INPUT_FILE=xml_filename
```

or

```
create suggLink --KEYWORD=keyword --LINK_URL=url -i xml_filename
```

Parameters

keyword

Content of the <search:keyword> element in the XML document.

url

Content of the <search:linkUrl> element in the XML document.

xml_filename

Path to the XML document that defines the suggested link. See [suggLink](#) on page 2-105.

Example

This example creates a suggested link for the `oracle` keyword from the description in `sugglinks.xml`:

```
SES>create suggLink --KEYWORD=oracle --LINK_URL=http://www.oracle.com --INPUT_
FILE=sugglinks.xml
```

The object "[keyword=oracle, linkUrl=http://www.oracle.com]" was successfully created.

create thesaurus

Creates a thesaurus from an XML description.

Syntax

```
create thesaurus --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create thesaurus -n object_name -i xml_filename
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that defines the object. See [thesaurus](#) on page 2-108.

Example

This example creates the DEFAULT thesaurus defined in thesaurus.xml:

```
SES>create thesaurus --NAME=DEFAULT --INPUT_FILE=thesaurus.xml
```

The object "[name=DEFAULT]" was successfully created.

createAll altWord

Creates all alternate words described in an XML file.

See Also

[create altWord](#)

Syntax

```
createAll altWord --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll altWord -i xml_filename [-d action]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [altWord](#) on page 2-26.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

Example

This example creates the three altWord objects defined in altwords.xml.

```
SES>createAll altWord --INPUT_FILE=altwords.xml
```

```
createAll operation succeeded for type "altWord".
```

```
3 object(s) with status CREATE_SUCCEEDED
```

The next example shows use of the --DUPE_METHOD option:

```
SES>createAll altWord --INPUT_FILE=altwords.xml
```

```
The object with key "[keyword=text, altKeyword=Oracle Text]" and type "altWord" already exists.
```

```
SES>createAll altWord --INPUT_FILE=altwords.xml --DUPE_METHOD=overwrite
```

```
createAll operation succeeded for type "altWord".
```

```
2 object(s) with status CREATE_SUCCEEDED
```

```
1 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll clusterTree

Creates all cluster trees described in an XML file.

See Also

[create clusterTree](#)

Syntax

```
createAll clusterTree --INPUT_FILE=xml_filename [--DUPE_METHOD=action [--IGNORE_INVALID_STATE=state]]
```

or

```
createAll clusterTree -i xml_filename [-d action [-s state]]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [clusterTree](#) on page 2-30.

action

Action to take when an object already exists:

- `error`: The createAll command fails with an error. (Default)
- `ignore`: The existing object description is kept.
- `overwrite`: The new description replaces the existing object description.

state

Controls whether processing continues when an object cannot be overwritten because it is in an invalid state. This option is used only when `--DUPE_METHOD=overwrite`.

- `true`: Continue processing with the next object.
- `false`: Stop processing with an error and roll back all changes. (Default)

Example

This command creates one of the two cluster trees described in `clustertree.xml`. The second cluster tree already exists.

```
SES>createAll clusterTree --INPUT_FILE=clustertree.xml --DUPE_METHOD=overwrite
```

```
createAll operation succeeded for type "clusterTree".
```

```
1 object(s) with status CREATE_SUCCEEDED
1 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll identityPlugin

Creates all identity plug-ins described in an XML file.

See Also

[create identityPlugin](#)

Syntax

```
createAll identityPlugin --INPUT_FILE=xml_filename [--DUPE_METHOD=action]  
[--ENCRYPT_KEY=key]
```

or

```
createAll identityPlugin -i xml_filename [-d action [-s state]] [-e key]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [identityPlugin](#) on page 2-38.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example attempts to create the identity plug-ins described in `identity.xml`, but it already exists:

```
SES>createAll identityPlugin --INPUT_FILE=identity.xml --DUPE_METHOD=ignore  
--ENCRYPT_KEY=key2encrypt
```

```
createAll operation succeeded for type "identityPlugin".
```

```
1 object(s) with status DUPLICATE_IGNORED
```

createAll proxyLogin

Creates all proxy log-ins described in an XML file.

See Also

[create proxyLogin](#)

Syntax

```
createAll proxyLogin --INPUT_FILE=xml_filename [--DUPE_METHOD=action] [--ENCRYPT_
KEY=key]
```

or

```
createAll proxyLogin -i xml_filename [-d action] [-e key]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [proxyLogin](#) on page 2-49.

action

Action to take when an object already exists:

- `error`: The createAll command fails with an error. (Default)
- `ignore`: The existing object description is kept.
- `overwrite`: The new description replaces the existing object description.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example creates all proxy log-ins described in proxy.xml. The status message shows that one of them already exists.

```
SES>createAll proxyLogin --INPUT_FILE=proxy.xml --DUPE_METHOD=ignore --ENCRYPT_
KEY=key2decrypt
```

```
createAll operation succeeded for type "proxyLogin".
```

```
2 object(s) with status CREATE_SUCCEEDED
1 object(s) with status DUPLICATE_IGNORED
```

createAll schedule

Creates all schedules described in an XML file.

See Also

[create schedule](#)

Syntax

```
createAll schedule --INPUT_FILE=xml_filename [--DUPE_METHOD=action [--IGNORE_INVALID_STATE=state]]
```

or

```
createAll schedule -i xml_filename [-d action [-s state]]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [schedule](#) on page 2-59.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

state

Controls whether processing continues when an object cannot be overwritten because it is in an invalid state. This option is used only when `--DUPE_METHOD=overwrite`.

- **true**: Continue processing with the next object.
- **false**: Stop processing with an error and roll back all changes. (Default)

Example

This example creates three schedules described in `schedule.xml`. Two of the schedules already exist.

```
SES>createAll schedule --INPUT_FILE=schedule.xml --DUPE_METHOD=ignore
```

```
createAll operation succeeded for type "schedule".
```

```
3 object(s) with status CREATE_SUCCEEDED
2 object(s) with status DUPLICATE_IGNORED
```

createAll searchAttr

Creates all custom search attributes described in an XML file.

See Also

[create searchAttr](#)

Syntax

```
createAll searchAttr --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll searchAttr -i xml_filename [-d action]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [searchAttr](#) on page 2-63.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

Example

This example creates three objects described in searchattr.xml. A fourth object already exists.

```
SES>createAll searchAttr --INPUT_FILE=searchattr.xml --DUPE_METHOD=ignore
```

```
createAll operation succeeded for type "searchAttr".
```

```
3 object(s) with status CREATE_SUCCEEDED  
1 object(s) with status DUPLICATE_IGNORED
```

createAll source

Creates all sources described in an XML file.

See Also

[create source](#)

Syntax

```
createAll source --INPUT_FILE=xml_filename [--DUPE_METHOD=action] [--ENCRYPT_
KEY=key]
```

or

```
createAll source -i xml_filename [-d action] [-e key]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [source](#) on page 2-68.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example creates one source defined in sources.xml. The second source already exists.

```
SES>createAll source --INPUT_FILE=sources.xml --DUPE_METHOD=ignore --ENCRYPT_
KEY=key2encrypt
```

```
createAll operation succeeded for type "source".
```

```
1 object(s) with status CREATE_SUCCEEDED
1 object(s) with status DUPLICATE_IGNORED
```

createAll sourceGroup

Creates all source groups described in an XML file.

See Also

[create sourceGroup](#)

Syntax

```
createAll sourceGroup --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll creatable_type -i xml_filename [-d action]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [sourceGroup](#) on page 2-93.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

Example

This example creates a source groups described in sourcegroups.xml. A second source group already exists.

```
SES>createAll sourceGroup --INPUT_FILE=sourcegroups.xml --DUPE_METHOD=ignore
```

```
createAll operation succeeded for type "sourceGroup".
```

```
1 object(s) with status CREATE_SUCCEEDED  
1 object(s) with status DUPLICATE_IGNORED
```

createAll sourceType

Creates all source types described in an XML file.

See Also

[create sourceType](#)

Syntax

```
createAll sourceType --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll sourceType -i xml_filename [-d action]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [sourceType](#) on page 2-95.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

Example

This command creates four source types:

```
SES>createAll sourceType --INPUT_FILE=sourcetypes.xml --DUPE_METHOD=ignore
```

```
createAll operation succeeded for type "sourceType".
```

```
4 object(s) with status CREATE_SUCCEEDED  
21 object(s) with status DUPLICATE_IGNORED
```

createAll storageArea

Creates all of the storage areas described in an XML document.

See Also

[create storageArea](#)

Syntax

```
createAll storageArea --INPUT_FILE=xml_filename [--DUPE_METHOD=action] [--IGNORE_INVALID_STATE=state]
```

or

```
createAll storageArea -i xml_filename [-d action] [-s state]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [clusterTree](#) on page 2-30.

action

Action to take when an object already exists:

- `error`: The `createAll` command fails with an error. (Default)
- `ignore`: The existing object description is kept.
- `overwrite`: The new description replaces the existing object description.

state

Controls whether processing continues when an object cannot be overwritten because it is in an invalid state. This option is used only when `--DUPE_METHOD=overwrite`.

- `true`: Continue processing with the next object.
- `false`: Stop processing with an error and roll back all changes. (Default)

Example

This example creates a storage area defined in a file named `storage.xml`:

```
SES>createAll storageArea --INPUT_FILE=storage.xml
```

```
createAll operation succeeded for type "storageArea".
```

```
1 object(s) with status CREATE_SUCCEEDED
```

createAll suggLink

Creates all suggested links described in an XML file.

See Also

[create suggLink](#)

Syntax

```
createAll suggLink --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll suggLink -i xml_filename [-d action]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See [suggLink](#) on page 2-105.

action

Action to take when an object already exists:

- **error**: The createAll command fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description.

Example

This example creates two suggested links from the descriptions in sugglinks.xml. A third object already exists.

```
SES>createAll suggLink --INPUT_FILE=sugglinks.xml --DUPE_METHOD=overwrite
```

```
createAll operation succeeded for type "suggLink".
```

```
2 object(s) with status CREATE_SUCCEEDED
1 object(s) with status DUPLICATE_OVERWRITTEN
```

deactivate clustering

Deactivates clustering.

Syntax

```
deactivate clustering
```

Example

This example deactivates clustering:

```
SES>deactivate clustering
```

The object "clustering" was successfully deactivated

deactivate clusterTree

Deactivates a cluster tree.

Syntax

```
deactivate clusterTree --NAME=object_name
```

or

```
deactivate clusterTree -n object_name
```

Parameters

object_name

Content of a <search:name> element in the XML document.

Example

This example deactivates a cluster tree named Topic Tree:

```
SES>deactivate clusterTree --NAME="Topic Tree"
```

The object "[name=Topic Tree]" was successfully deactivated.

deactivate identityPlugin

Deactivates an identity plug-in.

Syntax

```
deactivate identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class
```

Parameters

jar_filename

Content of a <search:jarFilePath> element in the XML document.

class

Content of a <search:managerClassName> element in the XML document.

Example

This example deactivates an identity plug-in:

```
SES>deactivate identityPlugin --JAR_FILE=OIDPlugins.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.oid.OIDPluginManager
```

The object "[jarFilePath=OIDPlugins.jar, managerClassName=oracle.search.plugin.security.identity.oid.OIDPluginManager]" was successfully deactivated.

deactivate indexOptimizer

Deactivates index optimization.

Syntax

```
deactivate indexOptimizer
```

Example

This example deactivates index optimization:

```
SES>deactivate indexOptimizer
```

The object "indexOptimizer" was successfully deactivated.

deactivate resultList

Deactivates the result list customizations.

Syntax

```
deactivate resultList
```

Example

This example deactivates result list customizations:

```
SES>deactivate resultList
```

The object "resultList" was successfully deactivated.

deactivate schedule

Deactivates a schedule.

Syntax

```
deactivate schedule --NAME=object_name
```

or

```
deactivate schedule -n object_name
```

Parameters

object_name

Content of a <search:name> element in the XML document.

Example

This example fails to deactivate a schedule because it is executing:

```
SES>deactivate schedule --NAME="Doc Library"
```

Operation "deactivate" cannot be performed on an object with type "schedule" in state "EXECUTING".

deactivate skinBundle

Deactivates a skin bundle.

Syntax

```
deactivate skinBundle --NAME=object_name
```

or

```
deactivate skinBundle -n object_name
```

Parameters

object_name

Content of a <search:name> element in the XML document.

Example

This example deactivate a skin bundle named Holiday:

```
SES>deactivate skinBundle --NAME=holiday
```

The object "[name=holiday]" was successfully deactivated.

deactivate spaceCalculator

Deactivates the space calculator.

Syntax

```
deactivate spaceCalculator
```

Example

This example deactivates the space calculator:

```
SES>deactivate spaceCalculator
```

The object "spaceCalculator" was successfully deactivated.

delete creatable_type

Deletes a creatable object.

See Also

[deleteAll creatable_type](#)
[deleteList creatable_type](#)

Syntax

```
delete creatable_type object_key
```

Parameters

creatable_type

A creatable type:

alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
thesaurus

object_key

Unique identifier of the object. See the object description in [Chapter 2, "Administration Object Types."](#)

Example

This example deletes the Doc Library schedule.

```
SES>delete schedule --NAME="Doc Library"
```

The object "[name=Doc Library]" was successfully deleted.

deleteAll *creatable_type*

Deletes all objects of a specified type.

Caution: This operation deletes all objects, including those supplied with Oracle SES. Use a *key_pattern* to limit the operation to those objects you want to delete, especially when deleting *identityPlugin*, *sourceType*, and *storageArea* objects.

See Also

[delete creatable_type](#)
[deleteList creatable_type](#)

Syntax

```
deleteAll creatable_type [--IGNORE_INVALID_STATE=state] [key_pattern]
```

or

```
deleteAll creatable_type [-s state] [key_pattern]
```

Parameters

creatable_type

A creatable type:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

state

Controls whether *clusterTree*, *identityPlugin*, *schedule*, or *skinBundle* objects are skipped because they are in an invalid state.

- `true`: Objects are deleted regardless of their state.
- `false`: Invalid objects are not deleted. (Default)

key_pattern

Object key that specifies a subset of objects to process in the format *key=value*. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches exactly one character. In a multibyte character set, it matches one byte.

Examples

This example deletes all `clusterTree` objects:

```
SES>deleteAll clusterTree
```

```
deleteAll operation succeeded for type "clusterTree".
```

```
1 object(s) with status DELETE_SUCCEEDED
```

The next example deletes two `sourceType` objects with the string `Documentum` in the name:

```
SES>deleteAll sourceType --NAME=%Documentum%
```

```
deleteAll operation succeeded for type "sourceType".
```

```
2 object(s) with status DELETE_SUCCEEDED
```

deleteList *creatable_type*

Deletes objects of a specified type that are listed in a text file.

See Also

`delete creatable_type`
`deleteAll creatable_type`

Syntax

```
deleteList creatable_type --KEYS_FILE=key_filename [--IGNORE_NOT_FOUND=action]
[--IGNORE_INVALID_STATE=state]
```

or

```
deleteList creatable_type -k key_filename [-f action] [-s state]
```

Parameters

creatable_type

A creatable type:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

key_filename

Path to a text file that identifies the objects to be deleted. Each line of the file contains an object key. For example:

```
--NAME=this_proxy
--NAME=that_proxy
```

action

Controls the resulting action when an object in the list does not exist.

- `true`: The object is skipped and processing continues.
- `false`: Processing stops with an error. (Default)

state

Controls whether `clusterTree`, `identityPlugin`, `schedule`, or `skinBundle` objects are skipped because they are in an invalid state.

- `true`: Objects are deleted regardless of their state.
- `false`: Invalid objects are not deleted. (Default)

Example

This example deletes the identity plug-ins listed in identity.lst:

```
SES>deleteList identityPlugin --KEYS_FILE=identity.lst --IGNORE_NOT_FOUND=true
```

```
deleteList operation succeeded for type "identityPlugin".
```

```
2 object(s) with status DELETE_SUCCEEDED
```

export *creatable_type*

Returns the XML description of an object.

See Also

`export universal_type`
`exportAll creatable_type`
`exportList creatable_type`

Syntax

```
export creatable_type object_key [--OUTPUT_FILE=output_file] [--ENCRYPT_KEY=key]
```

or

```
export creatable_type object_key [-o output_file] [-e key]
```

Parameters

creatable_type

A creatable type:

alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
task
thesaurus

object_key

Unique identifier of the object. See the object description in [Chapter 2, "Administration Object Types."](#)

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `export` command creates a separate file for each one and appends the host name and port number to the base name.

key

Encryption key for passwords in the XML description of `identityPlugin`, `proxyLogin`, and `source` objects. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example displays the XML for the `rac` alternative word.

```
SES>export altWord --KEYWORD=rac --ALT_KEYWORD="Real Application Clusters"

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:altWords>
    <search:altWord>
      <search:keyword>rac</search:keyword>
      <search:altKeyword>Real Application Clusters</search:altKeyword>
      <search:autoExpand>>false</search:autoExpand>
    </search:altWord>
  </search:altWords>
</search:config>
```

The next example creates a file named acme.xml containing the XML document for the acme skin bundle.

```
SES>export skinBundle --NAME=acme --OUTPUT_FILE=acme.xml
```

The object "[name=acme]" was successfully exported.

4 attachment(s) written to file.

export *universal_type*

Returns the XML description of an object.

See Also

[export creatable_type](#)

Syntax

```
export universal_type [--OUTPUT_FILE=output_file]
```

or

```
export universal_type [-o output_file]
```

Parameters

universal_type

A universal type:

```
clustering
crawlerSettings
index
indexOptimizer
partitionConfig
queryConfig
resultList
spaceCalculator
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `export` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example displays an XML document for a `crawlerSettings` object:

```
SES>export crawlerSettings
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:crawlerSettings>
    <search:numThreads>5</search:numThreads>
    <search:numProcessors>3</search:numProcessors>
    <search:crawlDepth haslimit="">
      <search:limit>2</search:limit>
    </search:crawlDepth>
    <search:languageDetection enabled="" />
    <search:defaultLanguage>en</search:defaultLanguage>
    <search:crawlTimeout>30</search:crawlTimeout>
    <search:maxDocumentSize>10</search:maxDocumentSize>
    <search:defaultCharSet>8859_1</search:defaultCharSet>
    <search:cacheDirectory>
      /home/oracle/dbs/ses111/cache/
    </search:cacheDirectory>
```

```
<search:preserveDocumentCache enabled="" />
<search:servicePipeline enabled="">
  <search:pipelineName>Default pipeline</search:pipelineName>
</search:servicePipeline>
<search:verboseLogging enabled="" />
<search:logDirectory>/home/oracle/dbs/ses111/log/</search:logDirectory>
<search:logLanguage>en-US</search:logLanguage>
</search:crawlerSettings>
</search:config>
```

exportAll creatable_type

Returns the XML descriptions of all objects of a specific type.

See Also

[export creatable_type](#)
[exportList creatable_type](#)

Syntax

```
exportAll creatable_type [key_pattern] [--OUTPUT_FILE=output_file] [--ENCRYPT_  
KEY=key]
```

or

```
exportAll creatable_type [key_pattern] [-o output_file] [-e key]
```

Parameters

creatable_type

A creatable object type:

```
alert  
altWord  
clusterTree  
identityPlugin  
proxyLogin  
schedule  
searchAttr  
skinBundle  
source  
sourceGroup  
sourceType  
storageArea  
suggLink
```

output_file

The name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `export` command creates a separate file for each one and appends the host name and port number to the base name.

key_pattern

An object key that specifies a subset of objects to process in the format *key=value*. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

key

The encryption key for passwords in the XML description of `identityPlugin`, `proxyLogin`, and `source` objects. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example displays all suggested links:

```
SES>exportAll suggLink
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:suggLinks>
    <search:suggLink>
      <search:keyword>database</search:keyword>
      <search:linkUrl>
        http://www.oracle.com/technology/products/database/oracle11g
      </search:linkUrl>
      <search:linkText>Oracle11g</search:linkText>
    </search:suggLink>
    <search:suggLink>
      <search:keyword>oracle</search:keyword>
      <search:linkUrl>http://www.oracle.com</search:linkUrl>
      <search:linkText>Oracle</search:linkText>
    </search:suggLink>
    <search:suggLink>
      <search:keyword>ses</search:keyword>
      <search:linkUrl>
        http://www.oracle.com/technology/products/oses/index.html
      </search:linkUrl>
      <search:linkText>Oracle SES</search:linkText>
    </search:suggLink>
  </search:suggLinks>
</search:config>
```

The next example uses a key pattern to find the suggested link for Oracle.

```
SES>exportAll suggLink --KEYWORD=ora%
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:suggLinks>
    <search:suggLink>
      <search:keyword>oracle</search:keyword>
      <search:linkUrl>http://www.oracle.com</search:linkUrl>
      <search:linkText>Oracle</search:linkText>
    </search:suggLink>
  </search:suggLinks>
</search:config>
```

exportList *creatable_type*

Returns the XML descriptions of a list of objects of the same type.

See Also

[export creatable_type](#)
[exportAll creatable_type](#)

Syntax

```
exportList creatable_type --KEYS_FILE=key_filename [--IGNORE_NOT_FOUND=action]
[--ENCRYPT_KEY=key]
```

or

```
exportList creatable_type -k key_filename [-f action] [-e key]
```

Parameters

creatable_type

A creatable object type:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

key_filename

Path to a text file that containing the keys that identify the objects to be deleted. The objects must be the same object type. Each line of the file identifies an object using this format:

```
key=value [key=value]
```

For example, these keys identify altWord objects:

```
--KEYWORD=oses --ALT_KEYWORD="Oracle Secure Enterprise Search"
--KEYWORD=rac --ALT_KEYWORD="Real Application Clusters"
--KEYWORD=oem --ALT_KEYWORD="Oracle Enterprise Manager"
```

action

Controls the resulting action when an object in the list does not exist:

- true: The object is skipped and processing continues.
- false: Processing stops with an error. (Default)

key

Encryption key for passwords in the XML description of identityPlugin, proxyLogin, and source objects. The key must be at least eight ASCII characters

long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Examples

This example exports the XML descriptions of the alternate words listed in `altwords.lst`:

```
SES>exportList altWord --KEYS_FILE=altwords.lst
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:altWords>
    <search:altWord>
      <search:keyword>oses</search:keyword>
      <search:altKeyword>Oracle Secure Enterprise Search</search:altKeyword>
      <search:autoExpand></search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>rac</search:keyword>
      <search:altKeyword>Real Application Clusters</search:altKeyword>
      <search:autoExpand>>false</search:autoExpand>
    </search:altWord>
  </search:altWords>
</search:config>
```

The next example shows use of the `--IGNORE_NOT_FOUND` option:

```
SES>exportList altWord --KEYS_FILE=altwords.lst
```

The object with key "[keyword=oem, altKeyword=Oracle Enterprise Manager]" and type "altWord" was not found.

```
SES>exportList altWord --KEYS_FILE=altwords.lst --IGNORE_NOT_FOUND=true
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:altWords>
    <search:altWord>
      <search:keyword>oses</search:keyword>
      <search:altKeyword>Oracle Secure Enterprise Search</search:altKeyword>
      <search:autoExpand></search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>rac</search:keyword>
      <search:altKeyword>Real Application Clusters</search:altKeyword>
      <search:autoExpand>>false</search:autoExpand>
    </search:altWord>
  </search:altWords>
</search:config>
```

getAllObjectKeys

Returns the object keys for the specified object type.

Syntax

```
getAllObjectKeys creatable_type --OUTPUT_FILE=output_file [key_pattern...
```

or

```
getAllObjectKeys creatable_type -o output_file [key_pattern...
```

Parameters

creatable_type

A creatable object type:

```
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
thesaurus
```

output_file

Name of a file in which the exported object keys stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getAllObjectKeys` command creates a separate file for each one and appends the host name and port number to the base name. You can use this file as input to the `deleteList`, `exportList`, and `getStateList` operations.

key_pattern

Object key that specifies a subset of objects to process in the format `key=value`. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

Examples

This example returns all suggested links:

```
SES>getAllObjectKeys suggLink

--KEYWORD=database --LINK_
URL=http://www.oracle.com/technology/products/database/oracle11g
--KEYWORD=oracle --LINK_URL=http://www.oracle.com
--KEYWORD=ses --LINK_URL=http://www.oracle.com/technology/products/oses/index.html
]
```

The next example returns only the suggested links with a keyword that begins with data:

```
SES>getAllObjectKeys suggLink --KEYWORD=data%
```

```
[--KEYWORD=database --LINK_  
URL=http://www.oracle.com/technology/products/database/oracle11g]
```

getAPIVersion

Displays the version number of the Oracle SES Administration API.

Syntax

```
getAPIVersion
```

Example

This example displays the current version of the Administration API.

```
SES>getAPIVersion
```

```
11.1.2.2.0
```

getAllStates clusterTree

Returns the current state of all cluster trees as an XML document.

See Also

[getState clusterTree](#)
[getStateList clusterTree](#)

Syntax

```
getAllStates clusterTree [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]  
[key_pattern...]
```

or

```
getAllStates clusterTree [-o output_file] [-l status] [key_pattern...]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getAllStates` command creates a separate file for each one and appends the host name and port number to the base name.

key_pattern

Object key that specifies a subset of objects to process in the format *key=value*. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

Example

This example returns the XML documents for all cluster trees. The XML shows that both cluster trees are currently active.

```
SES>getAllStates clusterTree
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<search:state productVersion="11.1.2.2.0"  
xmlns:search="http://xmlns.oracle.com/search">  
  <search:objectStates>  
    <search:objectState>  
      <search:objectType>clusterTree</search:objectType>  
      <search:objectKey>  
        <search:keyPairs>  
          <search:keyPair>  
            <search:name>name</search:name>  
            <search:value>Metadata Tree</search:value>  
          </search:keyPair>  
        </search:keyPairs>  
      </search:objectKey>  
      <search:stateProperties>  
        <search:stateProperty>
```



```
<search:propertyName>status</search:propertyName>
<search:propertyValues>
  <search:propertyValue>
    <search:value>ACTIVE</search:value>
  </search:propertyValue>
</search:propertyValues>
</search:stateProperty>
</search:stateProperties>
</search:objectState>
<search:objectState>
  <search:objectType>clusterTree</search:objectType>
  <search:objectKey>
    <search:keyPairs>
      <search:keyPair>
        <search:name>name</search:name>
        <search:value>Topic Tree</search:value>
      </search:keyPair>
    </search:keyPairs>
  </search:objectKey>
  <search:stateProperties>
    <search:stateProperty>
      <search:propertyName>status</search:propertyName>
      <search:propertyValues>
        <search:propertyValue>
          <search:value>ACTIVE</search:value>
        </search:propertyValue>
      </search:propertyValues>
    </search:stateProperty>
  </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getAllStates identityPlugin

Returns the current state of all identity plug-ins as an XML document.

See Also

[getState identityPlugin](#)
[getStateList identityPlugin](#)

Syntax

```
getAllStates identityPlugin [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
[key_pattern...]
```

or

```
getAllStates identityPlugin [-o output_file] [-l status] [key_pattern...]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getAllStates` command creates a separate file for each one and appends the host name and port number to the base name.

key_pattern

Object key that specifies a subset of objects to process in the format `key=value`. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

Example

This example returns the current state of all identity plug-ins:

```
SES>getAllStates identityPlugin
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>identityPlugin</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>jarFilePath</search:name>
            <search:value>OIDPlugins.jar</search:value>
          </search:keyPair>
          <search:keyPair>
            <search:name>managerClassName</search:name>
            <search:value>oracle.search.plugin.security.identity.oid.OIDPluginManager</search:
value>
```

```
        </search:keyPair>
    </search:keyPairs>
</search:objectKey>
<search:stateProperties>
    <search:stateProperty>
        <search:propertyName>status</search:propertyName>
        <search:propertyValues>
            <search:propertyValue>
                <search:value>ACTIVE</search:value>
            </search:propertyValue>
        </search:propertyValues>
    </search:stateProperty>
</search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getAllStates schedule

Returns the current state of all schedules as an XML document.

See Also

[getState schedule](#)
[getStateList schedule](#)

Syntax

```
getAllStates schedule [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=state_property]  
[key_pattern...]
```

or

```
getAllStates schedule [-o output_file] [-l state_property] [key_pattern...]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getAllStates` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

List of one or more state properties: `lastCrawled`, `logFilePath`, `nextCrawl`, `scheduleError`, or `status`. Separate multiple properties with commas. All state properties are returned by default.

key_pattern

Object key that specifies a subset of objects to process in the format `key=value`. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

Example

This example uses the percent (%) wildcard character to find the `Doc Library` schedule and requests two status properties. The status is currently `SCHEDULED` and the next crawl is `MANUAL`.

```
SES>getAllStates schedule --NAME=Doc% --PROPERTY_LIST=status,nextCrawl
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<search:state productVersion="11.1.2.2.0"  
xmlns:search="http://xmlns.oracle.com/search">  
  <search:objectStates>  
    <search:objectState>  
      <search:objectType>schedule</search:objectType>  
      <search:objectKey>  
        <search:keyPairs>  
          <search:keyPair>
```

```
        <search:name>name</search:name>
        <search:value>Doc Library</search:value>
    </search:keyPair>
</search:keyPairs>
</search:objectKey>
<search:stateProperties>
  <search:stateProperty>
    <search:propertyName>status</search:propertyName>
    <search:propertyValues>
      <search:propertyValue>
        <search:value>SCHEDULED</search:value>
      </search:propertyValue>
    </search:propertyValues>
  </search:stateProperty>
  <search:stateProperty>
    <search:propertyName>nextCrawl</search:propertyName>
    <search:propertyValues>
      <search:propertyValue>
        <search:value>MANUAL</search:value>
      </search:propertyValue>
    </search:propertyValues>
  </search:stateProperty>
</search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getAllStates skinBundle

Returns the current state of all skin bundles as an XML document.

See Also

[getState skinBundle](#)
[getStateList skinBundle](#)

Syntax

```
getAllStates skinBundle [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=state_
property] [key_pattern...]
```

or

```
getAllStates skinBundle [-o output_file] [-l state_property] [key_pattern...]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getAllStates` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

List of one or more state properties: `lastModifiedDate` or `status`. Separate multiple properties with commas. All state properties are returned by default.

key_pattern

Object key that specifies a subset of objects to process in the format `key=value`. The value can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches one character. In a multibyte character set, it matches one byte.

Example

This example shows that the `example` skin bundle is active, and the `acme` skin bundle is inactive:

```
SES>getAllStates skinBundle
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>skinBundle</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>acme</search:value>
```

```
        </search:keyPair>
    </search:keyPairs>
</search:objectKey>
<search:stateProperties>
    <search:stateProperty>
        <search:propertyName>status</search:propertyName>
        <search:propertyValues>
            <search:propertyValue>
                <search:value>INACTIVE</search:value>
            </search:propertyValue>
        </search:propertyValues>
    </search:stateProperty>
</search:stateProperties>
</search:objectState>
<search:objectState>
    <search:objectType>skinBundle</search:objectType>
    <search:objectKey>
        <search:keyPairs>
            <search:keyPair>
                <search:name>name</search:name>
                <search:value>example</search:value>
            </search:keyPair>
        </search:keyPairs>
    </search:objectKey>
    <search:stateProperties>
        <search:stateProperty>
            <search:propertyName>status</search:propertyName>
            <search:propertyValues>
                <search:propertyValue>
                    <search:value>ACTIVE</search:value>
                </search:propertyValue>
            </search:propertyValues>
        </search:stateProperty>
    </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getState clustering

Returns the current state of clustering as an XML document.

Syntax

```
getState clustering [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState clustering [-o output_file] [-l status]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example shows the clustering is currently active:

```
SES>getState clustering
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>clustering</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>ACTIVE</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```


getState clusterTree

Returns the current state of a cluster tree as an XML document.

See Also

[getAllStates clusterTree](#)
[getStateList clusterTree](#)

Syntax

```
getState clusterTree --NAME=object_name [--OUTPUT_FILE=output_file] [--PROPERTY_
LIST=status]
```

or

```
getState clusterTree -n object_name [-o output_file] [-l status]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example shows that the Metadata Tree cluster tree is currently active.

```
SES>getState clusterTree --NAME=Metadata Tree
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>clusterTree</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Metadata Tree</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>ACTIVE</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```

```
        </search:stateProperties>  
    </search:objectState>  
</search:objectStates>  
</search:state>
```

getState identityPlugin

Returns the current state of an identity plug-in as an XML document.

See Also

[getAllStates identityPlugin](#)
[getStateList identityPlugin](#)

Syntax

```
getState identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class [--OUTPUT_
FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState identityPlugin --JAR_FILE=jar_filename --MANAGER_CLASS=class [-o output_
file] [-l status]
```

Parameters

jar_filename

Content of a `search:jarFilePath` element in the XML document.

class

Content of a `search:managerClassName` element in the XML document.

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example gets the current state of a particular identity plug-in:

```
SES>getState identityPlugin --JAR_FILE=OIDPlugins.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.oid.OIDPluginManager
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>identityPlugin</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>jarFilePath</search:name>
            <search:value>OIDPlugins.jar</search:value>
          </search:keyPair>
          <search:keyPair>
            <search:name>managerClassName</search:name>
            <search:value>
              oracle.search.plugin.security.identity.oid.OIDPluginManager
            </search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
    </search:objectState>
  </search:objectStates>
</search:state>
```

```
        </search:keyPairs>
    </search:objectKey>
    <search:stateProperties>
        <search:stateProperty>
            <search:propertyName>status</search:propertyName>
            <search:propertyValues>
                <search:propertyValue>
                    <search:value>ACTIVE</search:value>
                </search:propertyValue>
            </search:propertyValues>
        </search:stateProperty>
    </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getState index

Returns the percent of fragmentation of the index as an XML document.

Syntax

```
getState index [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=estimatedFragmentation]
```

or

```
getState index [-o output_file] [-l estimatedFragmentation]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example shows that the index has a current estimated fragmentation level of 52.98% and should be optimized:

```
SES>getState index
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>index</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>estimatedFragmentation</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>52.98</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```

getState indexOptimizer

Returns the current state of index optimization as an XML document.

Syntax

```
getState indexOptimizer [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=state_
properties]
```

or

```
getState indexOptimizer [-o output_file] [-l state_properties]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more of the `indexOptimizer` state properties. Separate multiple properties with commas. All state properties are returned by default. See [indexOptimizer](#) on page 2-44 for the list of state properties.

Example

This example shows that the index optimizer started at 5:44 PM GMT and is still executing:

```
SES>getState indexOptimizer
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>indexOptimizer</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>EXECUTING</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
        <search:stateProperty>
          <search:propertyName>startTime</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>Thu, 09 Jul 2009 17:44:43 GMT</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
```

```
</search:objectStates>  
</search:state>
```

getState resultList

Returns the current state of the result list customizations as an XML document.

Syntax

```
getState resultList [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState resultList [-o output_file]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example shows that the result list is currently active:

```
SES>getState resultList
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>resultList</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>ACTIVE</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```

getState schedule

Returns the current state of a schedule as an XML document.

See Also

[getAllStates schedule](#)
[getStateList schedule](#)

Syntax

```
getState schedule --NAME=object_name [--OUTPUT_FILE=output_file] [--PROPERTY_
LIST=state_properties]
```

or

```
getState schedule -n object_name [-o output_file] [-l state_properties]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more of these state properties: `lastCrawled`, `logFilePath`, `nextCrawl`, `scheduleError`, and `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example directs a schedule to a file named `mailing_schedule.xml`. The XML shows that the current status of the schedule is disabled.

```
SES>getState schedule --NAME="Mailing List Schedule" --OUTPUT_FILE=mailing_
schedule.xml
```

The `getState` operation succeeded.

```
$ more mailing_schedule.xml
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0" xmlns:search="http://xmlns.oracle.com/
search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>schedule</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Mailing list Schedule</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
    </search:objectState>
  </search:objectStates>
</search:state>
```

```
</search:objectKey>
<search:stateProperties>
  <search:stateProperty>
    <search:propertyName>status</search:propertyName>
    <search:propertyValues>
      <search:propertyValue>
        <search:value>DISABLED</search:value>
      </search:propertyValue>
    </search:propertyValues>
  </search:stateProperty>
</search:stateProperties>
</search:objectState>
</search:objectStates>
```

getState skinBundle

Returns the current state of a skin bundle as an XML document.

See Also

[getAllStates skinBundle](#)
[getStateList skinBundle](#)

Syntax

```
getState skinBundle --NAME=object_name [--OUTPUT_FILE=output_file] [--PROPERTY_
LIST=state_properties]
```

or

```
getState skinBundle -n object_name [-o output_file] [-l state_properties]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more of these state properties: `lastModifiedDate` and `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example shows that the Holiday skin bundle is currently inactive.

```
SES>getState skinBundle -n holiday
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>skinBundle</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Holiday</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
```

```
        <search:value>INACTIVE</search:value>
    </search:propertyValue>
</search:propertyValues>
</search:stateProperty>
</search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getState spaceCalculator

Returns the current state of the space calculator as an XML document.

Syntax

```
getState spaceCalculator [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=state_
properties]
```

or

```
getState spaceCalculator [-o output_file] [-l state_properties]
```

Parameters

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

List of one or more state properties: `endTime`, `error`, `startTime`, or `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example returns only the `endTime` state for the space calculator:

```
SES>getState spaceCalculator --PROPERTY_LIST=endTime
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>spaceCalculator</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>endTime</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>Mon, 12 Oct 2009 18:35:19 GMT</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```

getState task

Returns the current state of a task.

Syntax

```
getState task --NAME=object_name [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=state_properties]
```

or

```
getState task -n object_name [-o output_file] [-l state_properties]
```

Parameters

object_name

The name of a task:

```
resumeAllSpaceConsumingTasks  
stopAllSpaceConsumingTasks
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getState` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more of these state properties: `endTime`, `startTime`, or `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example shows that `resumeAllSpaceConsumingTasks` is currently scheduled. The other states are not set.

```
SES>getState task --NAME=resumeAllSpaceConsumingTasks
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<search:state productVersion="11.1.2.2.0"  
xmlns:search="http://xmlns.oracle.com/search">  
  <search:objectStates>  
    <search:objectState>  
      <search:objectType>task</search:objectType>  
      <search:objectKey>  
        <search:keyPairs>  
          <search:keyPair>  
            <search:name>name</search:name>  
            <search:value>resumeAllSpaceConsumingTasks</search:value>  
          </search:keyPair>  
        </search:keyPairs>  
      </search:objectKey>  
      <search:stateProperties>  
        <search:stateProperty>  
          <search:propertyName>status</search:propertyName>  
          <search:propertyValues>  
            <search:propertyValue>  
              <search:value>SCHEDULED</search:value>
```

```
        </search:propertyValue>
      </search:propertyValues>
    </search:stateProperty>
  </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>
```

getStateList clusterTree

Returns the current state of a list of objects of the same type.

See Also

[getState clusterTree](#)
[getAllStates clusterTree](#)

Syntax

```
getStateList clusterTree --KEYS_FILE=key_filename [--OUTPUT_FILE=output_file]
```

or

```
getStateList clusterTree -k key_filename [-o output_file]
```

Parameters

key_filename

Path to a text file that identifies the objects. Each line of the file contains an object key.
For example:

```
--NAME="Topic Tree"
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getStateList` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example displays the last crawl and next crawl dates for the schedules listed in `clustertrees.lst`:

```
SES>getStateList clusterTree --KEYS_FILE=clustertrees.lst
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>clusterTree</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Topic Tree</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>status</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>ACTIVE</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>
```



```
        </search:propertyValues>
      </search:stateProperty>
    </search:stateProperties>
  </search:objectState>
</search:objectStates>
</search:state>
```

getStateList identityPlugin

Returns the current state of a list of objects of the same type.

See Also

[getAllStates identityPlugin](#)
[getState identityPlugin](#)

Syntax

```
getStateList object_type --KEYS_FILE=key_filename [--OUTPUT_FILE=output_file]
```

or

```
getStateList object_type -k key_filename [-o output_file]
```

Parameters

key_filename

Path to a text file that identifies the objects. Each line of the file contains an object key. For example:

```
--JAR_FILE=OIDPlugins.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.oid.OIDPluginManager
--JAR_FILE=../oracleapplications/Siebel8Crawler.jar --MANAGER_
CLASS=oracle.search.plugin.security.identity.siebel.SiebelIdentityPluginMgr
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getStateList` command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example displays the last crawl and next crawl dates for the schedules listed in `identity.lst`:

```
SES>getStateList identityPlugin --KEYS_FILE=identity.lst
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>identityPlugin</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>jarFilePath</search:name>
            <search:value>OIDPlugins.jar</search:value>
          </search:keyPair>
          <search:keyPair>
            <search:name>managerClassName</search:name>
            <search:value>
              oracle.search.plugin.security.identity.oid.OIDPluginManager
            </search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
    </search:objectState>
  </search:objectStates>
</search:state>
```

```

        </search:keyPairs>
    </search:objectKey>
    <search:stateProperties>
        <search:stateProperty>
            <search:propertyName>status</search:propertyName>
            <search:propertyValues>
                <search:propertyValue>
                    <search:value>ACTIVE</search:value>
                </search:propertyValue>
            </search:propertyValues>
        </search:stateProperty>
    </search:stateProperties>
</search:objectState>
<search:objectState>
    <search:objectType>identityPlugin</search:objectType>
    <search:objectKey>
        <search:keyPairs>
            <search:keyPair>
                <search:name>jarFilePath</search:name>
                <search:value>
                    ../oracleapplications/Siebel8Crawler.jar
                </search:value>
            </search:keyPair>
            <search:keyPair>
                <search:name>managerClassName</search:name>
                <search:value>
                    oracle.search.plugin.security.identity.siebel.SiebelIdentityPluginMgr
                </search:value>
            </search:keyPair>
        </search:keyPairs>
    </search:objectKey>
    <search:stateProperties>
        <search:stateProperty>
            <search:propertyName>status</search:propertyName>
            <search:propertyValues>
                <search:propertyValue>
                    <search:value>INACTIVE</search:value>
                </search:propertyValue>
            </search:propertyValues>
        </search:stateProperty>
    </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>

```

getStateList schedule

Returns the current state of a list of schedules

See Also

[getAllStates schedule](#)
[getState schedule](#)

Syntax

```
getStateList schedule --KEYS_FILE=key_filename [--OUTPUT_FILE=output_file]
[--PROPERTY_LIST=state_properties]
```

or

```
getStateList schedule -k key_filename [-o output_file] [-l state_properties]
```

Parameters

key_filename

Path to a text file that identifies the objects. Each line of the file contains an object key. For example:

```
--NAME="Doc Library"
--NAME="SQL Script Library"
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getStateList` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more state properties: `lastCrawled`, `logFilePath`, `nextCrawl`, `scheduleError`, and `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example displays the last crawl and next crawl dates for the schedules listed in `schedules.lst`:

```
SES>getStateList schedule --KEYS_FILE=schedules.lst --PROPERTY_
LIST=lastCrawled,nextCrawl
```

```
<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>schedule</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Doc Library</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
    </search:objectState>
  </search:objectStates>
</search:state>
```

```

    </search:keyPairs>
  </search:objectKey>
<search:stateProperties>
  <search:stateProperty>
    <search:propertyName>nextCrawl</search:propertyName>
    <search:propertyValues>
      <search:propertyValue>
        <search:value>MANUAL</search:value>
      </search:propertyValue>
    </search:propertyValues>
  </search:stateProperty>
  <search:stateProperty>
    <search:propertyName>lastCrawled</search:propertyName>
    <search:propertyValues>
      <search:propertyValue key="Doc Library">
        <search:value>Thu, 21 May 2009 16:54:17 GMT</search:value>
      </search:propertyValue>
    </search:propertyValues>
  </search:stateProperty>
</search:stateProperties>
</search:objectState>
<search:objectState>
  <search:objectType>schedule</search:objectType>
  <search:objectKey>
    <search:keyPairs>
      <search:keyPair>
        <search:name>name</search:name>
        <search:value>SQL Script Library</search:value>
      </search:keyPair>
    </search:keyPairs>
  </search:objectKey>
  <search:stateProperties>
    <search:stateProperty>
      <search:propertyName>nextCrawl</search:propertyName>
      <search:propertyValues>
        <search:propertyValue>
          <search:value>MANUAL</search:value>
        </search:propertyValue>
      </search:propertyValues>
    </search:stateProperty>
    <search:stateProperty>
      <search:propertyName>lastCrawled</search:propertyName>
      <search:propertyValues>
        <search:propertyValue key="SQL Script Library">
          <search:value>Thu, 21 May 2009 16:57:18 GMT</search:value>
        </search:propertyValue>
      </search:propertyValues>
    </search:stateProperty>
  </search:stateProperties>
</search:objectState>
</search:objectStates>
</search:state>

```

getStateList skinBundle

Returns the current state of a list of skin bundles

See Also

[getAllStates skinBundle](#)
[getState skinBundle](#)

Syntax

```
getStateList skinBundle --KEYS_FILE=key_filename [--OUTPUT_FILE=output_file]
[--PROPERTY_LIST=state_properties]
```

or

```
getStateList skinBundle -k key_filename [-o output_file] [-l state_properties]
```

Parameters

key_filename

Path to a text file that identifies the objects. Each line of the file contains an object key. For example:

```
--NAME="Doc Library"
--NAME="SQL Script Library"
```

output_file

Name of a file in which the exported XML document is stored. You can specify a simple file name, a relative path, or a fully qualified path. When executing on multiple instances, the `getStateList` command creates a separate file for each one and appends the host name and port number to the base name.

state_properties

One or more state properties: `lastModifiedDate` and `status`. Separate multiple properties with commas. All state properties are returned by default.

Example

This example stores the current state of all skin bundles listed in `skins.lst` in a file named `skinstate.xml`.

```
SES>getStateList skinBundle --KEYS_FILE=skins.lst --OUTPUT_FILE=skinstate.xml
```

The `getStateList` operation succeeded.

start indexOptimizer

Starts index optimization

Syntax

```
start indexOptimizer]
```

Example

This example starts the index optimizer:

```
SES>start indexOptimizer
```

```
The "start" operation succeeded for "indexOptimizer".
```

start schedule

Starts a schedule.

Syntax

```
start schedule --NAME=object_name ]
```

or

```
start schedule -n object_name ]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

Example

This example starts a schedule named SQL Script Library:

```
SES>start schedule --NAME='SQL Script Library'
```

The "start" operation succeeded for "[name=SQL Script Library]".

start spaceCalculator

Starts the space calculator.

Syntax

```
start spaceCalculator
```

Example

This example starts the space calculator:

```
SES>start spaceCalculator
```

```
The "start" operation succeeded for "spaceCalculator".
```

start task

Starts a task.

Syntax

```
start task --NAME=object_name
```

Parameters

object_name

The name of a task:

```
resumeAllSpaceConsumingTasks  
stopAllSpaceConsumingTasks
```

Example

This example stops all space-consuming tasks:

```
SES>start task --NAME=stopAllSpaceConsumingTasks
```

The "start" operation succeeded for "[name=stopAllSpaceConsumingTasks]".

stop schedule

Stops a schedule from initiating a crawl, or stops the crawler if it has already started.

Syntax

```
stop schedule --NAME=object_name
```

or

```
stop schedule -n object_name
```

Parameters

object_name

Content of a <search:name> element in the XML document.

Example

This example stops a schedule named My File Source:

```
SES>stop schedule --NAME="My File Source"
```

The "stop" operation succeeded for "[name=My File Source]".

update alert

Updates an alert.

See Also

[updateAll alert](#)

"Disk Space Management: Quotas and Alerts" on page 2-12

Syntax

```
update alert --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_METHOD=method
```

or

```
update alert -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that configures the object. See [alert](#) on page 2-23.

method

Method to use when updating the properties of an object:

- `overwrite`: Replaces the existing property values with the new values.

Example

This example updates the status of `alert_1`:

```
SES>update alert --NAME=alert_1 --INPUT_FILE=alert_update.xml --UPDATE_
METHOD=overwrite
```

The object "[name=alert_1]" was successfully updated.

update altWord

Changes the properties of an alternate word from an XML file.

See Also

[updateAll altWord](#)

Syntax

```
update altWord --KEYWORD=keyword --ALT_KEYWORD=alt_keyword object_key --INPUT_
FILE=xml_filename --UPDATE_METHOD=method
```

or

```
update altWord --KEYWORD=keyword --ALT_KEYWORD=alt_keyword -i xml_filename -a
method
```

Parameters

keyword

Content of a `<search:keyword>` element in the XML document.

alt_keyword

Content of a `<search:altKeyword>` element in the XML document.

xml_filename

Path to the XML document that configures the object. See [altWord](#) on page 2-26.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

Example

This example overwrites the parameter settings for RAC:

```
SES>update altWord --KEYWORD=rac --ALT_KEYWORD="Real Application Clusters"
--INPUT_FILE=altwords.xml --UPDATE_METHOD=overwrite
```

The object "[keyword=rac, altKeyword=Real Application Clusters]" was successfully updated.

update clustering

Sets one or more clustering properties from an XML description.

Syntax

```
update clustering --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update clustering -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [clustering](#) on page 2-28.

Example

This example updates the clustering properties:

```
SES>update clustering --INPUT_FILE=clustering.xml --UPDATE_METHOD=overwrite
```

The object "clustering" was successfully updated.

update clusterTree

Changes the properties of a cluster tree from an XML file.

See Also

[updateAll clusterTree](#)

Syntax

```
update clusterTree --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update clusterTree -n object_key -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [clusterTree](#) on page 2-30.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Example

This example overwrites the parameter settings for the Topic Tree cluster tree:

```
SES>update clusterTree --NAME=Topic Tree --UPDATE_METHOD=overwrite --INPUT_
FILE=clustertree.xml
```

The object "[name=Topic Tree]" was successfully updated.

update crawlerSettings

Sets one or more global crawler properties from an XML description.

Syntax

```
update crawlerSettings --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update crawlerSettings -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [crawlerSettings](#) on page 2-34.

Example

This example updates the crawler settings:

```
SES>update crawlerSettings --INPUT_FILE=crawler.xml --UPDATE_METHOD=overwrite
```

The object "crawlerSettings" was successfully updated.

update index

Sets one or more index properties from an XML description.

Syntax

```
update index --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update index -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [index](#) on page 2-42.

Example

This example updates the indexing settings:

```
SES>update index --INPUT_FILE=index.xml --UPDATE_METHOD=overwrite
```

The object "index" was successfully updated.

update indexOptimizer

Sets one or more index optimizer properties from an XML description.

Syntax

```
update indexOptimizer --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update indexOptimizer -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [indexOptimizer](#) on page 2-44.

Example

This example updates the index optimization settings:

```
SES>update indexOptimizer --INPUT_FILE=indexoptimizer.xml --UPDATE_  
METHOD=overwrite
```

The object "indexOptimizer" was successfully updated.

update partitionConfig

Changes the settings of the partition configuration parameters from an XML description. You can update the configuration only under these conditions:

- Immediately after installing Oracle SES.
- Before crawling any data source.
- After dropping all data sources.

When a partition rule is in use, you cannot modify it.

Syntax

```
update partitionConfig --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

```
update partitionConfig -a method -i xml_filename
```

Parameters

method

Controls the method used to update the properties of an object. Enter an update method:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [partitionConfig](#) on page 2-47.

Example

The following example updates the partitioning configuration:

```
SES>update partitionConfig --INPUT_FILE=part.xml --UPDATE_METHOD=overwrite
```

The object "partitionConfig" was successfully updated.

update proxyLogin

Changes the properties of a proxy log-in (federation entrusted entity) from an XML description.

See Also

[updateAll proxyLogin](#)

Syntax

```
update proxyLogin --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method [--ENCRYPT_KEY=key]
```

or

```
update proxyLogin -n object_name -i xml_filename -a method [-e key]
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [proxyLogin](#) on page 2-49.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example overwrites the parameter settings for a proxy:

```
SES>update proxyLogin --NAME=this_proxy --INPUT_FILE=proxy.xml --UPDATE_
METHOD=overwrite --ENCRYPT_KEY=key2encrypt
```

The object "[name=this_proxy]" was successfully updated.

update queryConfig

Sets one or more query configuration parameters from an XML description.

Syntax

```
update queryConfig --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update queryConfig -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [queryConfig](#) on page 2-51.

Example

This example updates the query settings:

```
SES>update queryConfig --INPUT_FILE=query.xml
```

The object "queryConfig" was successfully updated.

update resultList

Updates the list of search attributes that can be used for rendering the result list. However, these attributes appear in the result list only if the XSLT style sheet uses them.

Syntax

```
update resultList --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update resultList -a method -i xml_filename
```

Parameters

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

xml_filename

Path to the XML document that configures the object. See [resultList](#) on page 2-57.

Example

This example updates the result list properties:

```
SES>update resultList --UPDATE_METHOD=add --INPUT_FILE=resultlist.xml
```

The object "resultList" was successfully updated.

update schedule

Changes the properties of a schedule from an XML file.

See Also

[updateAll schedule](#)

Syntax

```
update schedule --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update creatable_type -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [schedule](#) on page 2-59.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Example

This example adds a source to the Doc Library schedule:

```
SES>update schedule --NAME="Doc Library" --UPDATE_METHOD=add --INPUT_
FILE=schedule.xml
```

The object "[name=Doc Library]" was successfully updated.

update searchAttr

Changes the properties of a search attribute from an XML file. You can add translations of search attribute names and LOVs to the default search attributes.

See Also

[updateAll searchAttr](#)

Syntax

```
update searchAttr --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update searchAttr -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [searchAttr](#) on page 2-63.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Example

This example overwrites the parameter settings for the Owner search attribute:

```
SES>update searchAttr --NAME=Owner --UPDATE_METHOD=overwrite --INPUT_
FILE=searchattrs.xml
```

The object "[name=Owner]" was successfully updated.

update skinBundle

Changes the properties of a skin bundle from an XML file.

See Also

["Search Interface Customization: Skin Bundles"](#) on page 2-13

Syntax

```
update skinBundle --NAME=object_name --INPUT_FILE=xml_filename [--ATTACHMENT_
LIST=list_filename] --UPDATE_METHOD=method
```

or

```
update skinBundle -n object_name -i xml_filename [-h list_filename] -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [skinBundle](#) on page 2-66.

list_filename

Path to the text file that lists the files in the skin bundle. This parameter is optional when the XML description does not update the files. See the Notes for ["create skinBundle"](#) on page 3-18.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Notes

After updating a skin bundle, restart the middle tier:

```
ORACLE_HOME/bin/searchctl restart
```

Example

This example updates the Holiday skin bundle.

```
SES>update skinBundle --NAME=holiday --INPUT_FILE=skins_update.xml --ATTACHMENT_
LIST=skins/resources.lst --UPDATE_METHOD=remove
```

The object "[name=holiday]" was successfully updated.

update source

Changes the properties of a source from an XML file.

See Also

[updateAll source](#)

Syntax

```
update source --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_METHOD=method
[--ENCRYPT_KEY=key]
```

or

```
update source -n object_name -i xml_filename -a method [-e key]
```

Parameters

object_name

Content of a `<search:name>` element in the XML document.

xml_filename

Path to the XML document that configures the object. See [source](#) on page 2-68.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

Example

This example updates the Doc Library source with the changes in sources.xml:

```
SES>update source --NAME="Doc Library" --UPDATE_METHOD=overwrite --INPUT_
FILE=sources.xml --ENCRYPT_KEY=key2encrypt
```

The object "[name=Doc Library]" was successfully updated.

update sourceGroup

Changes the properties of a source group from an XML file.

See Also

[updateAll sourceGroup](#)

Syntax

```
update sourceGroup --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update sourceGroup -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [sourceGroup](#) on page 2-93.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the Libraries source group:

```
SES>update sourceGroup --NAME=Libraries --UPDATE_METHOD=overwrite --INPUT_
FILE=sourcegroups.xml
```

The object "[name=Libraries]" was successfully updated.

update sourceType

Changes the properties of a source type from an XML file.

See Also

[updateAll sourceType](#)

Syntax

```
update sourceType --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update sourceType -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [sourceType](#) on page 2-95.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the definition of the EMC Documentum Content Server source type:

```
SES>update sourceType --NAME="EMC Documentum Content Server" --UPDATE_
METHOD=overwrite --INPUT_FILE=sourcetypes.xml
```

The object "[name=EMC Documentum Content Server]" was successfully updated.

update spaceCalculator

Sets one or more space calculator properties from an XML description.

Syntax

```
update spaceCalculator --INPUT_FILE=xml_filename --UPDATE_METHOD=method
```

or

```
update spaceCalculator -i xml_filename -a method
```

Parameters

xml_filename

Path to an XML document that contains parameter settings for the object. See [spaceCalculator](#) on page 2-99.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the space calculator schedule:

```
SES>update spaceCalculator --INPUT_FILE=space_update.xml --UPDATE_METHOD=overwrite
```

The object "spaceCalculator" was successfully updated.

update storageArea

Changes the properties of a storage area from an XML file.

Syntax

```
update storageArea --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update storageArea -n object_name -i xml_filename -a method
```

Parameters

object_name

Content of a <search:name> element in the XML document.

xml_filename

Path to the XML document that configures the object. See [storageArea](#) on page 2-102.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the parameter settings for the OES_ASSM2 storage area:

```
SES>update storageArea --NAME=OES_ASSM2 --UPDATE_METHOD=overwrite --INPUT_
FILE=storage.xml
```

The object "[name=OES_ASSM2]" was successfully updated.

update suggLink

Changes the properties of a suggested link from an XML file.

See Also

[updateAll suggLink](#)

Syntax

```
update suggLink --KEYWORD=keyword --LINK_URL=url --INPUT_FILE=xml_filename  
--UPDATE_METHOD=method
```

or

```
update suggLink --KEYWORD=keyword --LINK_URL=url -i xml_filename -a method
```

Parameters

keyword

Content of the search:keyword element in the XML document.

url

Content of the search:linkUrl element in the XML document.

xml_filename

Path to the XML document that configures the object. See [suggLink](#) on page 2-105.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the suggested link for oracle:

```
SES>update suggLink --KEYWORD=oracle --LINK_URL=http://www.oracle.com --UPDATE_  
METHOD=overwrite --INPUT_FILE=sugglinks.xml
```

The object "[keyword=oracle, linkUrl=http://www.oracle.com]" was successfully updated.

update thesaurus

Changes the properties of a thesaurus from an XML file.

Syntax

```
update thesaurus --NAME=DEFAULT --INPUT_FILE=xml_filename --UPDATE_METHOD=method
```

or

```
update thesaurus -n DEFAULT -i xml_filename -a method
```

Parameters

xml_filename

Path to the XML document that configures the object. See [thesaurus](#) on page 2-108.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

Example

This example updates the default thesaurus:

```
SES>update thesaurus --NAME=DEFAULT --UPDATE_METHOD=overwrite --INPUT_  
FILE=thesaurus_update.xml
```

The object "[name=DEFAULT]" was successfully updated.

updateAll alert

Updates all alerts.

See Also

[update alert](#)

"Disk Space Management: Quotas and Alerts" on page 2-12

Syntax

```
updateAll alert --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_
METHOD=action]
```

or

```
updateAll alert -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [alert](#) on page 2-23.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues.

Example

This example updates the status of all alerts described in alert_update.xml:

```
SES>updateAll alert --INPUT_FILE=alert_update.xml --UPDATE_METHOD=overwrite
```

```
updateAll operation succeeded for type "alert".
```

```
2 object(s) with status UPDATE_SUCCEEDED
```

updateAll altWord

Sets one or more parameters for all alternate words from an XML description.

See Also

[update altWord](#)

Syntax

```
updateAll altWord --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll altWord -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [altWord](#) on page 2-26.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues.

Example

This example updates the alternate word list by adding a new altWord object in altword2.xml:

```
SES>updateAll altWord --INPUT_FILE=altword2.xml --UPDATE_METHOD=overwrite --NOT_FOUND_METHOD=create
```

```
updateAll operation succeeded for type "altWord".
```

```
1 object(s) with status NOT_FOUND_CREATED
```

updateAll clusterTree

Sets one or more parameters for all cluster trees from an XML description.

See Also

[update clusterTree](#)

Syntax

```
updateAll clusterTree --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll clusterTree -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [clusterTree](#) on page 2-30.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates two cluster trees:

```
SES>updateAll clusterTree --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_FILE=clustertree2.xml
```

```
updateAll operation succeeded for type "clusterTree".
```

```
2 object(s) with status UPDATE_SUCCEEDED
```

updateAll proxyLogin

Sets one or more parameters for all proxy log-ins (federated trusted entities) from an XML description.

See Also

[update proxyLogin](#)

Syntax

```
updateAll proxyLogin --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--ENCRYPT_KEY=key] [--NOT_FOUND_METHOD=action]
```

or

```
updateAll proxyLogin -i xml_filename -a method [-e key] [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [proxyLogin](#) on page 2-49.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates all proxy log-ins described in proxy.xml:

```
SES>updateAll proxyLogin --INPUT_FILE=proxy.xml --UPDATE_METHOD=overwrite  
--ENCRYPT_KEY=key2encrypt
```

```
updateAll operation succeeded for type "proxyLogin".
```

```
3 object(s) with status UPDATE_SUCCEEDED
```

updateAll schedule

Sets one or more parameters for all schedules from an XML description.

See Also

[update schedule](#)

Syntax

```
updateAll schedule --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll schedule -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [schedule](#) on page 2-59.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates the schedules with the XML document in `schedule_rev1.xml`:

```
SES>updateAll schedule --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_FILE=schedule_rev1.xml
```

```
updateAll operation succeeded for type "schedule".
```

```
1 object(s) with status UPDATE_SUCCEEDED
```

updateAll searchAttr

Updates all search attributes from the descriptions in an XML file. You can add LOVs to the default search attributes.

See Also

[update searchAttr](#)

Syntax

```
updateAll searchAttr --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll searchAttr -i xml_filename -a method [-e key] [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [searchAttr](#) on page 2-63.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates a search attribute using the description in `searchattrs_update.xml`:

```
SES>updateAll searchAttr --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_FILE=searchattrs_update.xml
```

```
updateAll operation succeeded for type "searchAttr".
```

```
1 object(s) with status UPDATE_SUCCEEDED
```

updateAll source

Sets one or more parameters for all sources from an XML description.

See Also

[update source](#)

Syntax

```
updateAll source --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_
METHOD=action] [--ENCRYPT_KEY=key]
```

or

```
updateAll source -i xml_filename -a method [-t action] [-e key]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [source](#) on page 2-68.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

key

Encryption key for passwords in *xml_filename*. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid. If you omit this option, you are prompted for the key.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example creates a new source described in *sources_update.xml*:

```
SES>updateAll source --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_
FILE=sources_update.xml --ENCRYPT_KEY=key2encrypt
```

```
updateAll operation succeeded for type "source".
```

```
1 object(s) with status NOT_FOUND_CREATED
```

updateAll sourceGroup

Sets one or more parameters for all source groups from an XML description.

See Also

[update sourceGroup](#)

Syntax

```
updateAll sourceGroup --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll sourceGroup -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [sourceGroup](#) on page 2-93.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates two source groups:

```
SES>updateAll sourceGroup --UPDATE_METHOD=add --NOT_FOUND_METHOD=ignore --INPUT_FILE=sourcegroups_update.xml
```

```
updateAll operation succeeded for type "sourceGroup".
```

```
2 object(s) with status UPDATE_SUCCEEDED
2 object(s) with status NOT_FOUND_IGNORED
```

updateAll sourceType

Sets one or more parameters for all source types from an XML description.

See Also

[update sourceType](#)

Syntax

```
updateAll sourceType --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll sourceType -i xml_filename -a method [-e key] [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [sourceType](#) on page 2-95.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues.

Example

This example updates one source type and creates another source type:

```
SES>updateAll sourceType --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_FILE=sourcetypes_update.xml
```

```
updateAll operation succeeded for type "sourceType".
```

```
1 object(s) with status UPDATE_SUCCEEDED
1 object(s) with status NOT_FOUND_CREATED
```

updateAll storageArea

Changes the properties of all storage areas from an XML file.

Syntax

```
updateAll storageArea --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll storageArea -n object_name -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [storageArea](#) on page 2-102.

method

Method to use when updating the properties of an object:

- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted in the XML file.
- **remove**: Removes existing properties that are defined in the XML file.
- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example creates a storage area and updates the parameter settings of another:

```
SES> updateAll storageArea --UPDATE_METHOD=remove --NOT_FOUND_METHOD=create  
--INPUT_FILE=storage.xml
```

```
updateAll operation succeeded for type "storageArea".
```

```
1 object(s) with status UPDATE_SUCCEEDED  
1 object(s) with status NOT_FOUND_CREATED
```

updateAll suggLink

Updates all suggested links from an XML description.

See Also

[update suggLink](#)

Syntax

```
updateAll suggLink --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_METHOD=action]
```

or

```
updateAll suggLink -i xml_filename -a method [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See [suggLink](#) on page 2-105.

method

Method to use when updating the properties of an object:

- **overwrite**: Replaces the existing property values with the new values.

action

Action to take when an object does not exist:

- **create**: A new object is created from the XML description and processing continues.
- **error**: The command fails with an error. (Default)
- **ignore**: The new description is ignored and processing continues

Example

This example updates one suggested link and creates two more:

```
SES>updateAll suggLink --UPDATE_METHOD=overwrite --NOT_FOUND_METHOD=create  
--INPUT_FILE=sugglinks_update.xml
```

```
updateAll operation succeeded for type "suggLink".
```

```
1 object(s) with status UPDATE_SUCCEEDED  
2 object(s) with status NOT_FOUND_CREATED
```

Web Service Operations

This chapter describes the SOAP messages for the Oracle SES Web service. It contains this topic:

- [Alphabetic List of Web Service Operations](#)

Alphabetic List of Web Service Operations

A C D E G L S U

A

[activate](#)

C

[create](#)
[createAll](#)

D

[deactivate](#)
[delete](#)
[deleteAll](#)
[deleteList](#)

E

[export](#)
[exportAll](#)
[exportList](#)

G

[getAllObjectKeys](#)
[getAPIVersion](#)
[getAllStates](#)
[getState](#)
[getStateList](#)

L

[login](#)
[logout](#)

S

[start](#)
[stop](#)

U

[update](#)
[updateAll](#)

activate

Activates an administrative object. Objects that control the periodic execution of an activity must also be started: [indexOptimizer](#), [schedule](#), and [task](#).

See Also

[start](#)

SOAP Message

```
<activate xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
          <keyValue>
            <objectXML xmlns="">
              <decryptionKey xmlns="">
                <credentials xmlns="">
                  <password>
                    <userName>
              <locale xmlns="">
```

Element Descriptions

<activate xmlns="http://search.oracle.com/Admin">

Describes activation of an administrative object. It contains these elements:

```
<objectType>
<objectKey>
<objectXML>
<decryptionKey>
<credentials>
<locale>
```

<objectType xmlns="">

Contains one of these object types:

```
clustering
clusterTree
identityPlugin
indexOptimizer
resultList
schedule
skinBundle
spaceCalculator
```

<objectKey xmlns="">

Describes the object key for a creatable object type:

```
clusterTree
identityPlugin
schedule
```

This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

<keyName>
<keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as Website Schedule.

<objectXML xmlns="">

Contains an XML document that describes the object. See [Chapter 2, "Administration Object Types."](#)

Use escape codes for these symbols in the embedded XML:

Symbol	Escape Code
<	<
>	>
"	"

<decryptionKey xmlns="">

Contains a decryption key for `identityPlugin` objects.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

<password>
<userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault

Example

This example activates clustering:

```
<activate xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">clustering</objectType>
```


</activate>

This is the service response for a successful operation:

```
<ns2:activateResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

create

Creates an administrative object from an XML description.

See Also

[createAll](#)

SOAP Message

```
<create xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      </adminKeyPairs>
    </objectKey>
    <objectXML xmlns="">
    <decryptionKey xmlns="">
    <credentials xmlns="">
      <password>
      <userName>
    </credentials>
    <attachments xmlns="">
      <resourceName>
      <resourcePath>
    </attachments>
    <locale xmlns="">
```

Element Descriptions

<create xmlns="http://search.oracle.com/Admin">

Describes creation of an administrative object. It contains these elements:

```
<creatableType>
<objectKey>
<objectXML>
<decryptionKey>
<credentials>
<attachments>
<locale>
```

<creatableType xmlns="">

Contains a creatable type:

```
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
thesaurus
```

<objectKey xmlns="">

Describes the object key for the object. This element contains an `<adminKeyPairs>` element.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as `name`.

<keyValue>

Contains the value that uniquely describes the object, such as `Website Schedule`.

<objectXML xmlns="">

Contains an XML document that describes the object. See [Chapter 2, "Administration Object Types."](#)

Use escape codes for these symbols in the embedded XML:

Symbol	Escape Code
<	<
>	>
"	"

<decryptionKey xmlns="">

Contains a decryption key for these object types:

```
identityPlugin
proxyLogin
source
```

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for `<userName>`.

<userName>

Contains the user name of the Oracle SES administrator.

<attachments xmlns="">

Describes a set of files composing a `skinBundle`. It contains these elements:

```
<resourceName>
<resourcePath>
```

<resourceName>

Contains the relative path of the file within the skin bundle, such as `assets/images/logo.gif`

<resourcePath>

Contains the full local path to the attachment file, such as
/home/user/skins/acme/assets/images/logo.gif.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault

Example

This example creates an alternate word pair.

```
<create xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">altWord</creatableType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>keyword</keyName>
      <keyValue>oses</keyValue>
    </adminKeyPairs>
    <adminKeyPairs>
      <keyName>altKeyword</keyName>
      <keyValue>Oracle Secure Enterprise Search</keyValue>
    </adminKeyPairs>
  </objectKey>
  <objectXML xmlns="">

    <!-- XML object description appears here-->

  </objectXML>
</create>
```

This is the service response for a successful operation:

```
<ns2:createResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

createAll

Creates all of the administrative objects of a specified type in an XML description.

See Also

[create](#)

SOAP Message

```
<createAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectXML xmlns="">
      <decryptionKey xmlns="">
        <credentials xmlns="">
          <password>
            <userName>
          <controls xmlns="">
            <controlName>
              <controlValue>
            <locale xmlns="">
```

Element Descriptions

<createAll xmlns="http://search.oracle.com/Admin">

Describes creation of one or more administrative objects. It contains these elements:

```
<creatableType>
<objectXML>
<decryptionKey>
<credentials>
<attachments>
<controls>
<locale>
```

<creatableType xmlns="">

Contains a creatable type:

```
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
source
sourceGroup
sourceType
storageArea
suggLink
```

<objectXML xmlns="">

Contains an XML document that describes the objects. See [Chapter 2, "Administration Object Types."](#)

Use escape codes for these symbols in the embedded XML:

Symbol	Escape Code
<	<
>	>
"	"

<decryptionKey xmlns="">

Contains a decryption key for these object types:

```
identityPlugin  
proxyLogin  
source
```

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>  
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<controls xmlns="">

Specifies an operation control. It contains these elements:

```
<controlName>  
<controlValue>
```

<controlName>

A control name:

- **DUPE_METHOD**: Controls the action to take when an object already exists. For all creatable types.
- **IGNORE_INVALID_STATE**: Controls whether processing continues when an object cannot be overwritten because it is in an invalid state. This control is used only when **DUPE_METHOD** is set to **overwrite**. For `clusterTree`, `identityPlugin`, and `schedule` only.

<controlValue>

For **DUPE_METHOD**:

- **error**: The operation fails with an error. (Default)
- **ignore**: The existing object description is kept.
- **overwrite**: The new description replaces the existing object description

For **IGNORE_INVALID_STATE**:

- **true**: Continue processing with the next object.
- **false**: Stop processing with an error and roll back all changes. (Default)

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example creates all of the alternate word pairs defined in the object XML (omitted):

```
<createAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">altWord</creatableType>
  <objectXML xmlns=""

    <!-- XML object descriptions appear here-->

  </objectXML>
  <controls xmlns="">
    <controlName>DUPE_METHOD</controlName>
    <controlValue>overwrite</controlValue>
  </controls>
  <credentials xmlns="">
    <password>password</password>
    <userName>egsys</userName>
  </credentials>
</createAll>
```

The response shows that three alternate word pairs were created successfully.

```
<ns2:createAllResponse xmlns:ns2="http://search.oracle.com/Admin">
  <statusList>
    <objectKey>
      <adminKeyPairs>
        <keyName>keyword</keyName>
        <keyValue>oses</keyValue>
      </adminKeyPairs>
      <adminKeyPairs>
        <keyName>altKeyword</keyName>
        <keyValue>Oracle Secure Enterprise Search</keyValue>
      </adminKeyPairs>
    </objectKey>
    <objectType>altWord</objectType>
    <statusCode>CREATE_SUCCEEDED</statusCode>
  </statusList>
  <statusList>
    <objectKey>
      <adminKeyPairs>
        <keyName>keyword</keyName>
        <keyValue>text</keyValue>
      </adminKeyPairs>
      <adminKeyPairs>
```

```
        <keyName>altKeyword</keyName>
        <keyValue>Oracle Text</keyValue>
    </adminKeyPairs>
</objectKey>
<objectType>altWord</objectType>
<statusCode>CREATE_SUCCEEDED</statusCode>
</statusList>
<statusList>
<objectKey>
    <adminKeyPairs>
        <keyName>keyword</keyName>
        <keyValue>rac</keyValue>
    </adminKeyPairs>
    <adminKeyPairs>
        <keyName>altKeyword</keyName>
        <keyValue>Real Application Clusters</keyValue>
    </adminKeyPairs>
</objectKey>
<objectType>altWord</objectType>
<statusCode>CREATE_SUCCEEDED</statusCode>
</statusList>
</ns2:createAllResponse>
```

deactivate

Deactivates an administrative object.

SOAP Message

```
<deactivate xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      <credentials xmlns="">
        <password>
        <userName>
      <locale xmlns="">
```

Element Descriptions

<deactivate xmlns="http://search.oracle.com/Admin">

Describes activation of an administrative object. It contains these elements:

```
<objectType>
<objectKey>
<credentials>
<locale>
```

<objectType xmlns="">

Contains one of these object types:

```
clustering
clusterTree
identityPlugin
indexOptimizer
resultList
schedule
skinBundle
spaceCalculator
```

<objectKey xmlns="">

Describes the object key for a creatable object type. See [Chapter 2, "Administration Object Types,"](#) for format of the object keys for these object types:

```
clusterTree
identityPlugin
schedule
```

This element contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value of <keyName>, such as Website Schedule for the name of a schedule.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example deactivates the Oracle Doc Library schedule. Any error messages are returned in Italian.

```
<deactivate xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">schedule</objectType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>Oracle Doc Library</keyValue>
    </adminKeyPairs>
  </objectKey>
  <locale xmlns="">it</locale>
</deactivate>
```

This is the service response for a successful operation:

```
<ns2:deactivateResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

delete

Deletes an administrative object.

See Also

[deleteAll](#), [deleteList](#)

SOAP Message

```
<delete xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      </adminKeyPairs>
    </objectKey>
    <credentials xmlns="">
      <password>
      <userName>
    </credentials>
    <locale xmlns="">
  </creatableType>
</delete>
```

Element Descriptions

<delete xmlns="http://search.oracle.com/Admin">

Describes deletion of an administrative object. It contains these elements:

- creatableType
- objectKey
- credentials
- locale

<creatableType xmlns="">

Contains one of these creatable types:

- alert
- altWord
- clusterTree
- identityPlugin
- proxyLogin
- schedule
- searchAttr
- skinBundle
- source
- sourceGroup
- sourceType
- storageArea
- suggLink
- thesaurus

<objectKey xmlns="">

Describes the object key for the object. This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

- <keyName>
- <keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as Website Schedule.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

<password>

<userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault

Example

This example deletes a proxy log-in named `this_proxy`:

```
<delete xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">proxyLogin</creatableType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>this_proxy</keyValue>
    </adminKeyPairs>
  </objectKey>
</delete>
```

This is the service response for a successful operation:

```
<ns2:deleteResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

deleteAll

Deletes all of the administrative objects of a specified type.

See Also

[delete](#), [deleteList](#)

SOAP Message

```
<deleteAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKeyPattern xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      <credentials xmlns="">
        <password>
        <userName>
      <controls xmlns="">
        <controlName>
        <controlValue>
      <locale xmlns="">
```

Element Descriptions

<deleteAll xmlns="http://search.oracle.com/Admin">

Describes deletion of administrative objects. It contains these elements:

```
<creatableType>
<objectKeyPattern>
<credentials>
<controls>
<locale>
```

<creatableType xmlns="">

Contains one of these creatable types:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

<objectKeyPattern xmlns="">

An object key that matches the objects to process. It contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

<keyName>
<keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains a case-sensitive value that matches the object key of one or more objects. It can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches exactly one character. In a multibyte character set, it matches one byte.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

<password>
<userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<controls xmlns="">

Describes an operation control. It contains these elements:

<controlName>
<controlValue>

<controlName>

Contains these control:

IGNORE_INVALID_STATE: Controls whether processing continues when an object cannot be deleted because it is in an invalid state. For `clusterTree`, `identityPlugin`, and `schedule` only.

<controlValue>

For IGNORE_INVALID_STATE:

- `true`: Continue processing with the next object.
- `false`: Stop processing with an error and roll back all changes. (Default)

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault

```

DependentObjectFault
InvalidInputFault
InvalidStateFault

```

Example

This example deletes all schedules that match the string My%.

```

<deleteAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">schedule</creatableType>
  <objectKeyPattern xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>My%</keyValue>
    </adminKeyPairs>
  </objectKeyPattern>
</deleteAll>

```

This response confirms that the schedule named My Mail was successfully deleted:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:deleteAllResponse xmlns:ns2="http://search.oracle.com/Admin">
      <statusList>
        <objectKey>
          <adminKeyPairs>
            <keyName>name</keyName>
            <keyValue>My Mail</keyValue>
          </adminKeyPairs>
        </objectKey>
        <objectType>schedule</objectType>
        <statusCode>DELETE_SUCCEEDED</statusCode>
      </statusList>
    </ns2:deleteAllResponse>
  </S:Body>
</S:Envelope>

```

deleteList

Deletes a list of administrative objects of a specified type.

See Also

[delete](#), [deleteAll](#)

SOAP Message

```
<deleteList xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKeys xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      </adminKeyPairs>
    </objectKeys>
    <credentials xmlns="">
      <password>
      <userName>
    </credentials>
    <controls xmlns="">
      <controlName>
      <controlValue>
    </controls>
    <locale xmlns="">
```

Element Descriptions

<deleteList xmlns="http://search.oracle.com/Admin">

Describes the deletion of one or more objects. It contains these elements:

```
<creatableType>
<objectKeys>
<credentials>
<controls>
<locale>
```

<creatableType xmlns="">

Contains one of these creatable types:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

<objectKeys xmlns="">

Describes the key for an object. One or more <objectKeys> elements compose the list of objects to delete. This element contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

<keyName>
<keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as `name`.

<keyValue>

Contains the value that uniquely describes the object, such as `Website Schedule`.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

<password>
<userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<controls xmlns="">

Describes an operation control. It contains these elements:

<controlName>
<controlValue>

<controlName>

`IGNORE_NOT_FOUND`: Controls the resulting action when an object in the list does not exist.

`IGNORE_INVALID_STATE`: Controls whether processing continues when an object cannot be deleted because it is in an invalid state. For `identityPlugin` and `schedule` only.

<controlValue>

For `IGNORE_NOT_FOUND`:

- `true`: The object is skipped and processing continues.
- `false`: Processing stops with an error. (Default)

For `IGNORE_INVALID_STATE`:

- `true`: Continue processing with the next object.
- `false`: Stop processing with an error and roll back all changes. (Default)

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
 CreatableAdminObjectFault
 DependentObjectFault
 InvalidInputFault
 InvalidStateFault

Example

This example deletes a list of three proxy log-ins:

```
<deleteList xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">proxyLogin</creatableType>
  <objectKeys>
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>this_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>some_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <objectKeys>
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>that_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <controls xmlns="">
    <controlName>IGNORE_NOT_FOUND</controlName>
    <controlValue>true</controlValue>
  </controls>
</deleteList>
```

The service response indicates that `this_proxy` and `that_proxy` were deleted successfully. Processing continued after `some_proxy` was not found.

```
<ns2:deleteListResponse xmlns:ns2="http://search.oracle.com/Admin">
  <statusList>
    <objectKey>
      <adminKeyPairs>
        <keyName>name</keyName>
        <keyValue>this_proxy</keyValue>
      </adminKeyPairs>
    </objectKey>
    <objectType>proxyLogin</objectType>
    <statusCode>DELETE_SUCCEEDED</statusCode>
  </statusList>
  <statusList>
    <objectKey>
      <adminKeyPairs>
        <keyName>name</keyName>
        <keyValue>some_proxy</keyValue>
      </adminKeyPairs>
    </objectKey>
    <objectType>proxyLogin</objectType>
    <statusCode>NOT_FOUND_IGNORED</statusCode>
  </statusList>
```

```
<statusList>
  <objectKey>
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>that_proxy</keyValue>
    </adminKeyPairs>
  </objectKey>
  <objectType>proxyLogin</objectType>
  <statusCode>DELETE_SUCCEEDED</statusCode>
</statusList>
</ns2:deleteListResponse>
```

export

Returns the XML description of an object.

See Also

[exportAll](#), [exportList](#)

SOAP Message

```
<export xmlns="http://search.oracle.com/Admin"
  <objectType xmlns="">
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>
      <keyValue>
    <encryptionKey xmlns="">
    <credentials xmlns="">
      <password>
      <userName>
    <locale xmlns="">
```

Element Descriptions

<export xmlns="http://search.oracle.com/Admin"

Describes the export of an administrative object. See [Table 1–2, "Universal Objects"](#) and [Table 1–3, "Creatable Object Types"](#). This element contains these child elements:

```
<objectType>
<objectKey>
<encryptionKey>
<credentials>
<locale>
```

<objectType xmlns="">

Contains an administration object type. See [Table 1–2, "Universal Objects"](#) and [Table 1–3, "Creatable Object Types"](#).

<objectKey xmlns="">

Describes the object key for the object. This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as `name`.

<keyValue>

Contains the value that uniquely describes the object, such as `Website Schedule`.

<encryptionKey xmlns="">

The encryption key for passwords in the XML description of `identityPlugin`, `proxyLogin`, and `source` objects. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example exports the XML description of this_proxy proxy log-in:

```
<export xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">proxyLogin</objectType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>this_proxy</keyValue>
    </adminKeyPairs>
  </objectKey>
  <encryptionKey xmlns="">key2encrypt</encryptionKey>
  <credentials xmlns="">
    <password>password</password>
    <userName>eqsys</userName>
  </credentials>
</export>
```

The service response contains the XML definition:

```
<ns2:exportResponse xmlns:ns2="http://search.oracle.com/Admin">
  <objectOutput>
    <objectXML>

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:proxyLogins>
    <search:proxyLogin>
      <search:name>this_proxy</search:name>
      <search:password encrypted="true">128b6b430...</search:password>
```

```
        <search:useIdentityPlugin>>false</search:useIdentityPlugin>
    </search:proxyLogin>
</search:proxyLogins>
</search:config>

    </objectXML>
</objectOutput>
</ns2:exportResponse>
```

exportAll

Returns the XML descriptions of all objects of a specified type.

See Also

[export](#), [exportList](#)

SOAP Message

```
<exportAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKeyPattern xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      <encryptionKey xmlns="">
      <credentials xmlns="">
        <password>
        <userName>
      <locale xmlns="">
```

Element Descriptions

<exportAll xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<creatableType>
<objectKeyPattern>
<encryptionKey>
<credentials>
<locale>
```

<creatableType xmlns="">

Contains one of these creatable types:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

<objectKeyPattern xmlns="">

An object key that matches the objects to process. It contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains a value that matches the object key of one or more objects. It can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches exactly one character. In a multibyte character set, it matches one byte.

<encryptionKey xmlns="">

The encryption key for passwords in the XML description of `identityPlugin`, `proxyLogin`, and `source` objects. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>  
<userName>
```

<password>

Contains the password for `<userName>`.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault  
CreatableAdminObjectFault  
DependentObjectFault  
InvalidInputFault  
InvalidStateFault
```

Example

This example exports the XML descriptions of all alternate word pairs:

```
<exportAll xmlns="http://search.oracle.com/Admin">  
  <creatableType xmlns="">altWord</creatableType>  
  <credentials xmlns="">  
    <password>password</password>  
    <userName>eqsys</userName>  
  </credentials>  
</exportAll>
```

The service response contains the XML definitions of three alternate word pairs.


```
<ns2:exportAllResponse xmlns:ns2="http://search.oracle.com/Admin">
  <objectOutput>
    <objectXML>

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:altWords>
    <search:altWord>
      <search:keyword>oses</search:keyword>
      <search:altKeyword>Oracle Secure Enterprise Search</search:altKeyword>
      <search:autoExpand>true</search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>rac</search:keyword>
      <search:altKeyword>Real Application Clusters</search:altKeyword>
      <search:autoExpand>false</search:autoExpand>
    </search:altWord>
    <search:altWord>
      <search:keyword>text</search:keyword>
      <search:altKeyword>Oracle Text</search:altKeyword>
      <search:autoExpand>false</search:autoExpand>
    </search:altWord>
  </search:altWords>
</search:config>

    </objectXML>
  </objectOutput>
</ns2:exportAllResponse>
```

exportList

Returns the XML descriptions of a list of objects of a specified type.

See Also

[export](#), [exportAll](#)

SOAP Message

```
<exportList xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <objectKeys xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      <encryptionKey xmlns="">
    <credentials xmlns="">
      <password>
      <userName>
    <controls xmlns="">
      <controlName>
      <controlValue>
    <locale xmlns="">
```

Element Descriptions

<exportList xmlns="http://search.oracle.com/Admin">

Describes a list of objects for export. It contains these elements:

```
<creatableType>
<objectKeys>
<encryptionKey>
<credentials>
<controls>
<locale>
```

<creatableType xmlns="">

Contains a creatable object type:

```
alert
altWord
clusterTree
identityPlugin
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggLink
```

<objectKeys xmlns="">

Describes the key for an object. One or more <objectKeys> elements compose the list of objects to export. This element contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

<keyName>
<keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as "Website Schedule".

<encryptionKey xmlns="">

The encryption key for passwords in the XML description of `identityPlugin`, `proxyLogin`, and `source` objects. The key must be at least eight ASCII characters long and include at least one letter and one number. Multibyte characters are invalid.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

<password>
<userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<controls xmlns="">

Describes an operation control. It contains these elements:

<controlName>
<controlValue>

<controlName>

IGNORE_NOT_FOUND: Controls the resulting action when an object in the list does not exist.

<controlValue>

For IGNORE_NOT_FOUND:

- true: The object is skipped and processing continues.
- false: Processing stops with an error. (Default)

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault

InvalidStateFault

Example

This example requests the XML descriptions of three proxy log-ins: `this_proxy`, `some_proxy`, and `that_proxy`.

```
<exportList xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">proxyLogin</creatableType>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>this_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>some_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>that_proxy</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <encryptionKey xmlns="">key2encrypt</encryptionKey>
  <controls xmlns="">
    <controlName>IGNORE_NOT_FOUND</controlName>
    <controlValue>true</controlValue>
  </controls>
</exportList>
```

The following response provides the XML descriptions of `this_proxy` and `that_proxy`. It ignored the request for `some_proxy`, which did not exist.

```
<ns2:exportListResponse xmlns:ns2="http://search.oracle.com/Admin">
  <objectOutput>
    <objectXML><?xml version="1.0" encoding="UTF-8"?>
      <search:config productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
        <search:proxyLogins>
          <search:proxyLogin>
            <search:name>this_proxy</search:name>
            <search:password encrypted="true">128b6b43...</search:password>
            <search:useIdentityPlugin>false</search:useIdentityPlugin>
          </search:proxyLogin>
          <search:proxyLogin>
            <search:name>that_proxy</search:name>
            <search:password encrypted="true">a625ca28...</search:password>
            <search:useIdentityPlugin>false</search:useIdentityPlugin>
          </search:proxyLogin>
        </search:proxyLogins>
      </search:config>
    </objectXML>
  </objectOutput>
</ns2:exportListResponse>
```

getAllObjectKeys

Returns the object keys for a specified object type.

SOAP Message

```
<getAllObjectKeys xmlns="http://search.oracle.com/Admin">>
  <creatableType xmlns="">
  <objectKeyPattern xmlns="">
    <adminKeyPairs>
      <keyName>
      <keyValue>
    <credentials xmlns="">
      <password>
      <userName>
    <locale xmlns="">
```

Element Descriptions

<getAllObjectKeys xmlns="http://search.oracle.com/Admin">>

Contains these elements:

```
<creatableType>
<objectKeyPattern>
<credentials>
<locale>
```

<creatableType xmlns="">

Contains a creatable object type. See [Table 1-3, "Creatable Object Types"](#).

<objectKeyPattern xmlns="">

An object key that matches the objects to process. It contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains a value that matches the object key of one or more objects. It can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches exactly one character. In a multibyte character set, it matches one byte.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>  
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

Example

This example requests all suggested links with a keyword that begins with data:

```
<getAllObjectKeys xmlns="http://search.oracle.com/Admin">  
  <creatableType xmlns=""> suggLink</creatableType>  
  <objectKeyPattern xmlns="">  
    <adminKeyPairs>  
      <keyName>keyword</keyName>  
      <keyValue>data%</keyValue>  
    </adminKeyPairs>  
  </objectKeyPattern>  
</getAllObjectKeys>
```

The service response provides the object keys for the suggested link with the keyword database:

```
<ns2:getAllObjectKeysResponse xmlns:ns2="http://search.oracle.com/Admin">  
  <objectKeyList>  
    <adminKeyPairs>  
      <keyName>keyword</keyName>  
      <keyValue>database</keyValue>  
    </adminKeyPairs>  
    <adminKeyPairs>  
      <keyName>linkUrl</keyName>  
      <keyValue>  
        http://www.oracle.com/technology/products/database/oracle11g  
      </keyValue>  
    </adminKeyPairs>  
  </objectKeyList>  
</ns2:getAllObjectKeysResponse>
```

getAPIVersion

Returns the version number of the Oracle SES Administration API.

SOAP Message

```
<getAPIVersion xmlns="http://search.oracle.com/Admin">
  <credentials xmlns="">
    <password>
    <userName>
  </credentials>
  <locale xmlns="">
```

Element Descriptions

<getAPIVersion xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<credentials>
<locale>
```

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
InvalidInputFault
```

Example

This example requests the version number of the API:

```
<getAPIVersion xmlns="http://search.oracle.com/Admin"/>
```

The service response contains the version number:

```
<ns2:getAPIVersionResponse xmlns:ns2="http://search.oracle.com/Admin">
  <return>11.1.2.2.0</return>
</ns2:getAPIVersionResponse>

<ns2:getAPIVersionResponse xmlns:ns2="http://search.oracle.com/Admin">
```

```
<version>11.1.2.2.0</version>  
</ns2:getAPIVersionResponse>
```


getAllStates

Returns the current state of all objects of a specified type as an XML document.

See Also

[getState](#), [getStateList](#)

SOAP Message

```
<getAllStates xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">
    <stateProperties xmlns="">
      <objectKeyPattern xmlns="">
        <adminKeyPairs>
          <keyName>
            <keyValue>
          </keyValue>
        </adminKeyPairs>
      </objectKeyPattern>
      <credentials xmlns="">
        <password>
        <userName>
      </credentials>
      <locale xmlns="">
    </stateProperties>
  </creatableType>
</getAllStates>
```

Element Descriptions

<getAllStates xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<creatableType>
<stateProperties>
<objectKeyPattern>
<credentials>
<locale>
```

<creatableType xmlns="">

Contains one of these creatable types:

```
clusterTree
identityPlugin
schedule
skinBundle
```

<stateProperties xmlns="">

Contains a state property of the object. The response is limited to the properties listed in the `<stateProperties>` elements. By default, the response contains the current state of all properties. See [Table 1-4, "Administrative Objects With State Properties"](#).

<objectKeyPattern xmlns="">

An object key that matches the objects to process. It contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains a value that matches the object key of one or more objects. It can include wildcard characters:

- A percent sign (%) matches zero or more characters. In a multibyte character set, it matches zero or more bytes.
- An underscore (_) matches exactly one character. In a multibyte character set, it matches one byte.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example requests the status and next crawl time for schedules with a name beginning with Ora. Any error messages are returned in Spanish.

```
<getAllStates xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">schedule</creatableType>
  <stateProperties xmlns="">status</stateProperties>
  <stateProperties xmlns="">nextCrawl</stateProperties>
  <objectKeyPattern xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>Ora%</keyValue>
    </adminKeyPairs>
  </objectKeyPattern>
  <locale xmlns="">es</locale>
</getAllStates>
```

The service response describes a schedule named Oracle Doc Library. Its status is SCHEDULED, and the next crawl is Thu, 15 Oct 2009 08:00:00 GMT.

```
<ns2:getAllStatesResponse xmlns:ns2="http://search.oracle.com/Admin">
```

```
<objectStateXML>
  <objectXML>

  <?xml version="1.0" encoding="UTF-8"?>
  <search:state productVersion="11.1.2.2.0"
  xmlns:search="http://xmlns.oracle.com/search">
    <search:objectStates>
      <search:objectState>
        <search:objectType>schedule</search:objectType>
        <search:objectKey>
          <search:keyPairs>
            <search:keyPair>
              <search:name>name</search:name>
              <search:value>Oracle Doc Library</search:value>
            </search:keyPair>
          </search:keyPairs>
        </search:objectKey>
        <search:stateProperties>
          <search:stateProperty>
            <search:propertyName>status</search:propertyName>
            <search:propertyValues>
              <search:propertyValue>
                <search:value>SCHEDULED</search:value>
              </search:propertyValue>
            </search:propertyValues>
          </search:stateProperty>
          <search:stateProperty>
            <search:propertyName>nextCrawl</search:propertyName>
            <search:propertyValues>
              <search:propertyValue>
                <search:value>Thu, 15 Oct 2009 08:00:00 GMT</search:value>
              </search:propertyValue>
            </search:propertyValues>
          </search:stateProperty>
        </search:stateProperties>
      </search:objectState>
    </search:objectStates>
  </search:state>

  </objectXML>
</objectStateXML>
</ns2:getAllStatesResponse>
```

getState

Returns the current state of an object as an XML document.

See Also

[getAllStates](#), [getStateList](#)

SOAP Message

```
<getState xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      <stateProperties xmlns="">
      <credentials xmlns="">
        <password>
        <userName>
      <locale xmlns="">
```

Element Descriptions

<getState xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<objectType>
<objectKey>
<stateProperties>
<credentials>
<locale>
```

<objectType xmlns="">

Contains one of these object types:

```
clustering
clusterTree
identityPlugin
index
indexOptimizer
resultList
schedule
skinBundle
spaceCalculator
task
```

<objectKey xmlns="">

Describes the object key for a creatable object. This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as `Website Schedule`.

<stateProperties xmlns="">

Contains a state property of the object. The response is limited to the properties listed in the `<stateProperties>` elements. By default, the response contains the current state of all properties. See [Table 1–4, "Administrative Objects With State Properties"](#).

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
```

```
<userName>
```

<password>

Contains the password for `<userName>`.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2–2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example requests the current state of the index:

```
<getState xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">index</objectType>
</getState>
```

The service response shows that the index has an estimated fragmentation of 1.5%.

```
<ns2:getStateResponse xmlns:ns2="http://search.oracle.com/Admin">
  <objectStateXML>
    <objectXML>

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>index</search:objectType>
      <search:stateProperties>
        <search:stateProperty>
```

```
        <search:propertyName>estimatedFragmentation</search:propertyName>
        <search:propertyValues>
          <search:propertyValue>
            <search:value>1.5</search:value>
          </search:propertyValue>
        </search:propertyValues>
      </search:stateProperty>
    </search:stateProperties>
  </search:objectState>
</search:objectStates>
</search:state>

</objectXML>
</objectStateXML>
</ns2:getStateResponse>
```

getStateList

Returns the current state of a list of objects of the same type.

See Also

[getAllStates](#), [getState](#)

SOAP Message

```
<getStateList xmlns="http://search.oracle.com/Admin">
  <creatableType>
  <objectKeys>
    <adminKeyPairs>
      <keyName>
      <keyValue>
    <stateProperties>
  <credentials>
    <password>
    <userName>
  <locale xmlns="">
```

Element Descriptions

<getStateList xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<creatableType>
<objectKeys>
<stateProperties>
<credentials>
<locale>
```

<creatableType>

Contains one of these creatable types:

```
clusterTree
identityPlugin
schedule
skinBundle
```

<objectKeys>

Describes the key for an object. One or more `<objectKeys>` elements compose the list of objects. This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as `name`.

<keyValue>

Contains the value that uniquely describes the object, such as `Website Schedule`.

<stateProperties>

Contains a state property of the object. The response is limited to the properties listed in the <stateProperties> elements. By default, the response contains the current state of all properties. See [Table 1-4, "Administrative Objects With State Properties"](#).

<credentials>

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example requests the next crawl time for the Oracle Doc Library and SQL Script Library schedules:

```
<getStateList xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">schedule</creatableType>
  <stateProperties xmlns="">nextCrawl</stateProperties>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>Oracle Doc Library</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <objectKeys xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>SQL Script Library</keyValue>
    </adminKeyPairs>
  </objectKeys>
  <credentials xmlns="">
    <password>password</password>
    <userName>eqsys</userName>
  </credentials>
</getStateList>
```


The service response provides the next scheduled crawl times:

```

<ns2:getStateListResponse xmlns:ns2="http://search.oracle.com/Admin">
  <objectStateXML>
    <objectXML>

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.1.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:objectStates>
    <search:objectState>
      <search:objectType>schedule</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>Oracle Doc Library</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>nextCrawl</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>Thu, 15 Oct 2009 08:00:00 GMT</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
    <search:objectState>
      <search:objectType>schedule</search:objectType>
      <search:objectKey>
        <search:keyPairs>
          <search:keyPair>
            <search:name>name</search:name>
            <search:value>SQL Script Library</search:value>
          </search:keyPair>
        </search:keyPairs>
      </search:objectKey>
      <search:stateProperties>
        <search:stateProperty>
          <search:propertyName>nextCrawl</search:propertyName>
          <search:propertyValues>
            <search:propertyValue>
              <search:value>Wed, 07 Oct 2009 06:00:00 GMT</search:value>
            </search:propertyValue>
          </search:propertyValues>
        </search:stateProperty>
      </search:stateProperties>
    </search:objectState>
  </search:objectStates>
</search:state>

    </objectXML>
  </objectStateXML>
</ns2:getStateListResponse>

```

login

Provides the administrative credentials for a stateful session. While the client maintains the HTTP session, credentials are not required in subsequent SOAP messages.

SOAP Message

```
<login xmlns="http://search.oracle.com/Admin">
  <credentials xmlns="">
    <password>
    <userName>
  </credentials>
  <locale xmlns="">
```

Element Descriptions

<login xmlns="http://search.oracle.com/Admin">

Contains these elements:

```
<credentials>
<locale>
```

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. It contains these elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault

Example

This example provides the credentials for a stateful session:

```
<login xmlns="http://search.oracle.com/Admin">
  <credentials xmlns="">
    <password>password</password>
    <userName>eqsys</userName>
  </credentials>
</login>
```

This is the service response:

```
<ns2:loginResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

logout

Closes a stateful session. Credentials are required in subsequent SOAP messages.

SOAP Message

```
<logout xmlns="http://search.oracle.com/Admin" />
```

SOAP Faults

```
AdminAPIRuntimeFault
```

Example

This example closes a stateful session:

```
<logout xmlns="http://search.oracle.com/Admin" />
```

This is the service response for a successful operation:

```
<ns2:logoutResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

start

Starts an administrative object.

SOAP Message

```
<start xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
          <keyValue>
        <credentials xmlns="">
          <password>
          <userName>
        <locale xmlns="">
```

Element Descriptions

<start xmlns="http://search.oracle.com/Admin">

Describes creation of an administrative object. It contains these elements:

```
<objectType>
<objectKey>
<objectXML>
<credentials>
<locale>
```

<objectType xmlns="">

Contains one of these types:

```
indexOptimizer
schedule
spaceCalculator
task
```

<objectKey xmlns="">

Describes the object key for the object. This element contains one or more `<adminKeyPairs>` elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as "Website Schedule".

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
```

```
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example starts the index optimizer:

```
<start xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">indexOptimizer</objectType>
</start>
```

This is the service response for a successful operation:

```
<ns2:startResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

The next example starts the Oracle Doc Library schedule:

```
<start xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">schedule</objectType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>Oracle Doc Library</keyValue>
    </adminKeyPairs>
  </objectKey>
</start>
```

stop

Stops an administrative object.

SOAP Message

```
<stop xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
          <keyValue>
        <credentials xmlns="">
          <password>
          <userName>
        <locale xmlns="">
```

Element Descriptions

<stop xmlns="http://search.oracle.com/Admin">

Describes creation of an administrative object. It contains these elements:

```
<objectType>
<objectKey>
<objectXML>
<credentials>
<locale>
```

<objectType xmlns="">

Contains this object type:

```
schedule
```

<objectKey xmlns="">

Describes the object key for the object. This element contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

```
<keyName>
<keyValue>
```

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as "Website Schedule".

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Example

This example stops the Oracle Doc Library schedule:

```
<stop xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">schedule</objectType>
  <objectKey xmlns="">
    <adminKeyPairs>
      <keyName>name</keyName>
      <keyValue>Oracle Doc Library</keyValue>
    </adminKeyPairs>
  </objectKey>
</credentials>
</stop>
```

This is the service response for a successful operation:

```
<ns2:stopResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

update

Sets the properties of an administrative object.

See Also

[updateAll](#)

SOAP Message

```
<update xmlns="http://search.oracle.com/Admin">
  <objectType xmlns="">
    <objectKey xmlns="">
      <adminKeyPairs>
        <keyName>
        <keyValue>
      </adminKeyPairs>
    </objectKey>
    <objectXML xmlns="">
    <decryptionKey xmlns="">
    <credentials xmlns="">
      <password>
      <userName>
    </credentials>
    <attachments xmlns="">
      <resourceName>
      <resourcePath>
    </attachments>
    <controls xmlns="">
      <controlName>
      <controlValue>
    </controls>
    <locale xmlns="">
```

Element Descriptions

<update xmlns="http://search.oracle.com/Admin">

Describes the update of an administrative object. It contains these elements:

```
<objectType>
<objectKey>
<objectXML>
<objectXML>
<decryptionKey>
<credentials>
<attachments>
<controls>
<locale>
```

<objectType xmlns="">

Contains one of these types:

```
alert
altWord
clustering
clusterTree
crawlerSettings
index
indexOptimizer
partitionConfig
proxyLogin
queryConfig
resultList
schedule
```


searchAttr
 skinBundle
 source
 sourceGroup
 sourceType
 spaceCalculator
 storageArea
 suggLink
 thesaurus

<objectKey xmlns="">

Describes the object key for a creatable object. This element contains one or more <adminKeyPairs> elements.

<adminKeyPairs>

Contains these elements:

<keyName>
 <keyValue>

<keyName>

Contains the case-sensitive key name of the object type, such as name.

<keyValue>

Contains the value that uniquely describes the object, such as "Website Schedule".

<objectXML xmlns="">

Contains an XML document that describes the object. See [Chapter 2, "Administration Object Types."](#)

Use escape codes for these symbols in the embedded XML:

Symbol	Escape Code
<	<
>	>
"	"

<decryptionKey xmlns="">

Contains a decryption key for proxyLogin and source objects.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See "[Providing Credentials](#)" on page 1-5.

This element contains these child elements:

<password>
 <userName>

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<attachments xmlns="">

Describes a set of files composing a skinBundle. It contains these elements:

<resourceName>
<resourcePath>

<resourceName>

Contains the relative path of the file within the skin bundle, such as assets/images/logo.gif

<resourcePath>

Contains the full local path to the attachment file, such as /home/user/skins/holiday/assets/images/logo.gif.

<controls xmlns="">

Specifies an operation control. It contains these elements:

<controlName>
<controlValue>

<controlName>

UPDATE_METHOD: Controls the method used to update the properties of an object. (Required)

<controlValue>

For UPDATE_METHOD:

- **overwrite**: Replaces the existing property values with the new values.
- **add**: Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted from the XML document. For `clusterTree`, `resultList`, `schedule`, `searchAttr`, `source`, `sourceGroup`, and `sourceType`.
- **remove**: Removes existing properties. For `clusterTree`, `resultList`, `schedule`, `searchAttr`, `source`, `sourceGroup`, and `sourceType`.

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault

Notes

After updating a skin bundle, restart the middle tier:

```
ORACLE_HOME/bin/searchctl restart
```

Example

This example updates the crawler settings. The input XML document is omitted from the example.

```
<update xmlns="http://search.oracle.com/Admin">  
  <objectType xmlns="">crawlerSettings</objectType>
```

```
<objectXML xmlns="">
  <!-- XML object description appears here-->
</objectXML>
<controls xmlns="">
  <controlName>UPDATE_METHOD</controlName>
  <controlValue>overwrite</controlValue>
</controls>
</update>
```

The following is the service response to a successful update:

```
<ns2:updateResponse xmlns:ns2="http://search.oracle.com/Admin" />
```

updateAll

Updates all objects of a given type that are described in the XML.

See Also

[update](#)

SOAP Message

```
<updateAll xmlns="http://search.oracle.com/Admin">
  <creatableType>
    <objectXML xmlns="">
      <decryptionKey xmlns="">
        <credentials xmlns="">
          <password>
            <userName>
          <controls xmlns="">
            <controlName>
              <controlValue>
            <locale xmlns="">
```

Element Descriptions

<updateAll xmlns="http://search.oracle.com/Admin">

Describes creation of an administrative object. It contains these elements:

```
<creatableType>
<objectKey>
<objectXML>
<decryptionKey>
<credentials>
<attachments>
<controls>
<locale>
```

<creatableType>

Contains one of these types:

```
alert
altWord
clusterTree
proxyLogin
schedule
searchAttr
source
sourceGroup
sourceType
suggLink
```

<objectXML xmlns="">

Contains an XML document that describes the object. See [Chapter 2, "Administration Object Types."](#)

Use escape codes for these symbols in the embedded XML:

Symbol	Escape Code
<	<
>	>
"	"

<decryptionKey xmlns="">

Contains a decryption key for proxyLogin and source objects.

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See ["Providing Credentials"](#) on page 1-5.

This element contains these child elements:

```
<password>
<userName>
```

<password>

Contains the password for <userName>.

<userName>

Contains the user name of the Oracle SES administrator.

<controls xmlns="">

Specifies an operation control. It contains these elements:

```
<controlName>
<controlValue>
```

<controlName>

UPDATE_METHOD: Controls the method used to update the properties of an object.

NOT_FOUND_METHOD: Controls the action to take when an object does not exist.

IGNORE_INVALID_STATE: Controls whether processing continues when an object cannot be overwritten because it is in an invalid state. For clusterTree, schedule, and storageArea objects. This option is used only when UPDATE_METHOD is OVERWRITE.

<controlValue>

For UPDATE_METHOD:

- **overwrite:** Replaces the existing property values with the new values.
- **add:** Adds new properties and overwrites existing properties, but does not delete existing properties that are omitted from the XML document. For clusterTree, schedule, searchAttr, source, sourceGroup, sourceType, and storageArea.
- **remove:** Removes existing properties. For clusterTree, resultList, schedule, searchAttr, source, sourceGroup, sourceType, and storageArea.

For NOT_FOUND_METHOD:

- **create:** Creates a new object from the XML description and continue processing.
- **error:** Processing stops with an error. (Default)
- **ignore:** Ignore the new description and continue processing.

For IGNORE_INVALID_STATE:

- `true`: Continue processing with the next object.
- `false`: Stop processing with an error and roll back all changes. (Default)

<locale xmlns="">

Sets the language for error messages. See [Table 2-2, "Product Languages"](#) for a list of valid codes.

If you omit this element or enter an invalid code, then Oracle SES uses the default locale of the system where it is running.

SOAP Faults

```
AdminAPIRuntimeFault
CreatableAdminObjectFault
DependentObjectFault
InvalidInputFault
InvalidStateFault
```

Notes

After updating a skin bundle, restart the middle tier:

```
ORACLE_HOME/bin/searchctl restart
```

Example

This example updates all of the `altWord` objects described in the `<objectXML>` element. The content of this element is not shown.

```
<updateAll xmlns="http://search.oracle.com/Admin">
  <creatableType xmlns="">altWord</creatableType>
  <objectXML xmlns="">

    <!-- XML object descriptions appear here-->

  </objectXML>
  <controls xmlns="">
    <controlName>UPDATE_METHOD</controlName>
    <controlValue>OVERWRITE</controlValue>
  </controls>
  <controls xmlns="">
    <controlName>NOT_FOUND_METHOD</controlName>
    <controlValue>CREATE</controlValue>
  </controls>
</updateAll>
```

The service response shows that two alternate keywords were updated and one was created.

```
<ns2:updateAllResponse xmlns:ns2="http://search.oracle.com/Admin">
  <statusList>
    <objectKey>
      <adminKeyPairs>
        <keyName>keyword</keyName>
        <keyValue>oses</keyValue>
      </adminKeyPairs>
      <adminKeyPairs>
        <keyName>altKeyword</keyName>
        <keyValue>Oracle Secure Enterprise Search</keyValue>
```

```
        </adminKeyPairs>
    </objectKey>
    <objectType>altWord</objectType>
    <statusCode>UPDATE_SUCCEEDED</statusCode>
</statusList>
<statusList>
    <objectKey>
        <adminKeyPairs>
            <keyName>keyword</keyName>
            <keyValue>rac</keyValue>
        </adminKeyPairs>
        <adminKeyPairs>
            <keyName>altKeyword</keyName>
            <keyValue>Real Application Clusters</keyValue>
        </adminKeyPairs>
    </objectKey>
    <objectType>altWord</objectType>
    <statusCode>NOT_FOUND_CREATED</statusCode>
</statusList>
<statusList>
    <objectKey>
        <adminKeyPairs>
            <keyName>keyword</keyName>
            <keyValue>text</keyValue>
        </adminKeyPairs>
        <adminKeyPairs>
            <keyName>altKeyword</keyName>
            <keyValue>Oracle Text</keyValue>
        </adminKeyPairs>
    </objectKey>
    <objectType>altWord</objectType>
    <statusCode>UPDATE_SUCCEEDED</statusCode>
</statusList>
</ns2:updateAllResponse>
```

Java Example

This appendix contains a programming example of the Web Services Java client. It contains these topics:

- [Java Source Code Example](#)
- [Shell Script Example](#)

See Also:

- ["Using the Web Services Java Client"](#) on page 1-5
- *Oracle Secure Enterprise Search Java API Reference*

Java Source Code Example

The following source code is an example named `CreateWebSource.java`. If you want, you can copy and paste this example into a file on your Oracle SES host.

This example uses a stateless Administration API client to do the following:

1. Create a Web source named `web1`.
2. Export `web1` to show the full definition.
3. Create a source group named `Web` containing `web1`.
4. Create and start a schedule named `schedule1` for `web1`.
5. Print the status of `schedule1` after 30 seconds.

`CreateWebSource.java` obtains values for these variables from the command-line arguments:

- `webServiceURL`
- `userName`
- `password`
- `webSourceURL`

For more information about these arguments, see ["Shell Script Example"](#) on page A-5.

```
import oracle.search.admin.api.ws.client.AdminAPIRuntimeFault;
import oracle.search.admin.api.ws.client.AdminAPIRuntimeFault_Exception;
import oracle.search.admin.api.ws.client.AdminKeyPair;
import oracle.search.admin.api.ws.client.AdminPortType;
import oracle.search.admin.api.ws.client.AdminService;
import oracle.search.admin.api.ws.client.Credentials;
import oracle.search.admin.api.ws.client.ObjectKey;
import oracle.search.admin.api.ws.client.ObjectOutput;
```

```
import java.util.List;
import java.net.URL;

import javax.xml.ws.BindingProvider;
import javax.xml.namespace.QName;

public class CreateWebSource
{
    public static void main(String[] args) throws Exception
    {
        System.out.println( " " );

        try
        {
            if ( args == null || args.length != 4 )
            {
                System.out.println(
                    "Usage:\n CreateWebSource <webServiceURL> <userName> <password>
<webSourceURL>"
                );
            }
            else
            {
                // Get web service URL from command-line arguments
                String webServiceURL = args[0];
                System.out.println( "Using web service URL \"" + webServiceURL + "\"\n" );

                // Get username and password
                String userName = args[1];
                String password = args[2];

                // Get stateless web service client
                AdminPortType adminPort =
                    getStatelessWebServiceClient( webServiceURL );

                // Create Credentials object for operations
                Credentials credentials = new Credentials();
                credentials.setUserName( userName );
                credentials.setPassword( password );

                // 1. Create a simple web source
                String webSourceURL = args[3];
                String webSourceXML =
                "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
                "<search:config productVersion=\"11.1.2.2.0\" " +
                "xmlns:search=\"http://xmlns.oracle.com/search\">" +
                "  <search:sources>" +
                "    <search:webSource>" +
                "      <search:name>web1</search:name>" +
                "      <search:startingUrls>" +
                "        <search:startingUrl>" +
                "          <search:url>" + webSourceURL + "</search:url>" +
                "        </search:startingUrl>" +
                "      </search:startingUrls>" +
                "    </search:webSource>" +
                "  </search:sources>" +
                "</search:config>";

                adminPort.createAll(
```

```

        "source",
        webSourceXML,
        "password",
        credentials,
        null,
        null,
        "en"
    );

    // 2. Export all sources to show the full definition
    ObjectOutput oo = adminPort.exportAll(
        "source",
        null,
        "password",
        credentials,
        null,
        "en"
    );
    System.out.println("Web Source XML = \n" + oo.getObjectXML() );

    // 3. Create a source group for the source
    String sourceGroupXML =
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>\" +
"<search:config productVersion=\"11.1.2.2.0\"\"
xmlns:search=\"http://xmlns.oracle.com/search\">\" +
"  <search:sourceGroups>\" +
"    <search:sourceGroup>\" +
"      <search:name>Web</search:name>\" +
"      <search:assignedSources>\" +
"        <search:assignedSource>web1</search:assignedSource>\" +
"      </search:assignedSources>\" +
"    </search:sourceGroup>\" +
"  </search:sourceGroups>\" +
"</search:config>\";

    adminPort.createAll(
        "sourceGroup",
        sourceGroupXML,
        null,
        credentials,
        null,
        null,
        "en"
    );

    System.out.println("Created source group...");

    // 4. Create a schedule for the web source
    String scheduleXML =
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>\" +
"<search:config productVersion=\"11.1.2.2.0\"\"
xmlns:search=\"http://xmlns.oracle.com/search\">\" +
"  <search:schedules>\" +
"    <search:schedule>\" +
"      <search:name>schedule1</search:name>\" +
"      <search:crawlingMode>ACCEPT_ALL</search:crawlingMode>\" +
"      <search:recrawlPolicy>PROCESS_CHANGED</search:recrawlPolicy>\" +
"      <search:frequency>\" +
"        <search:manual/>\" +
"      </search:frequency>\" +

```

```
"    <search:assignedSources>" +
"    <search:assignedSource>web1</search:assignedSource>" +
"    </search:assignedSources>" +
"    </search:schedule>" +
"  </search:schedules>" +
"</search:config>";

    adminPort.createAll(
        "schedule",
        scheduleXML,
        null,
        credentials,
        null,
        null,
        "en"
    );

    System.out.println("Created schedule...");

    // 5. Start the schedule

    // Create object key for schedule name
    ObjectKey objectKey = new ObjectKey();
    AdminKeyPair keyPair = new AdminKeyPair();
    keyPair.setKeyName( "name" ); // schedules identified by name
    keyPair.setKeyValue( "schedule1" ); // schedule name
    objectKey.getAdminKeyPairs().add( keyPair );

    adminPort.start(
        "schedule",
        objectKey,
        null,
        null,
        credentials,
        null,
        null,
        "en"
    );

    System.out.println("Started schedule...");
    System.out.println("Waiting 30 seconds to get status...");
    Thread.sleep( 30000 );

    // 6. Use object key above to get schedule state
    oo = adminPort.getState(
        "schedule",
        objectKey,
        null, // request all state properties
        credentials,
        null,
        "en"
    );

    System.out.println("Schedule state XML = " + oo.getObjectXML() );
}
}
catch (AdminAPIRuntimeFault_Exception e)
{
    AdminAPIRuntimeFault runtimeFault = e.getFaultInfo();
    System.out.println("Caught AdminAPIRuntimeFault");
}
```

```

        System.out.println(" message      = " + runtimeFault.getMessage() );
        System.out.println(" errorCode    = " + runtimeFault.getErrorCode() );
        System.out.println(" causeMessage = " + runtimeFault.getCauseMessage() );
        System.out.println(" stackTrace  = " );
        e.printStackTrace( System.out );
        System.out.println(" causeStackTrace = \n" +
runtimeFault.getCauseStackTrace() );
    }
    catch (Throwable t)
    {
        System.out.println("Caught unexpected run-time exception");
        System.out.println(" message    = " + t.getMessage() );
        System.out.println(" stackTrace = " );
        t.printStackTrace( System.out );
    }
}

/**
 * Initializes and returns a stateless admin web service client.
 */
private static AdminPortType getStatelessWebServiceClient(
    String webServiceURL) throws Exception
{
    AdminService adminService = new AdminService(
        new URL( webServiceURL ),
        new QName(
            "http://search.oracle.com/Admin",
            "AdminService"
        )
    );

    return adminService.getAdmin();
}
}

```

Shell Script Example

This example uses a shell script (command file) named `compileAndRun.sh` to compile and run `CreateWebSource.java`. You can use JRockit, as shown here, or any JDK 6 that contains a JAX-WS 2.1 implementation.

```

#!/bin/sh

CLASSPATH=.:$ORACLE_HOME/search/lib/search_adminapi_wsclient.jar:$ORACLE_
BASE/jrockit_160_14_R27.6.5-32/jre/lib/rt.jar

# Compile
$ORACLE_BASE/jrockit_160_14_R27.6.5-32/bin/javac -cp $CLASSPATH
CreateWebSource.java

# Run
$ORACLE_BASE/jrockit_160_14_R27.6.5-32/jre/bin/java -cp $CLASSPATH CreateWebSource
$a

```

To run the script, include these arguments on the command line:

- **webServiceURL:** The Web Service URL for the Administration API in the following format. Replace `host:port` with the appropriate values.

```
http://host:port/search/api/admin/AdminService
```

- **userName:** The administrative user (eqsys).
- **password:** The password for the administrative user.
- **webSourceURL:** The starting URL for crawling the Web source.

This command creates a source from the example.com Web site:

```
sh compileAndRun.sh http://host:7777/search/api/admin/AdminService eqsys password  
http://example.com/index.htm
```

Error Messages

This appendix describes general error messages from the Administration API. The messages are grouped into these categories:

- [Invalid Input Errors](#)
- [Creatable Administration Object Errors](#)
- [Invalid State Errors](#)
- [Administration API Run-Time Errors](#)
- [Dependent Object Errors](#)

Note: A *propertyName* in a message uses XPath notation to identify the location of the property in the input XML.

See Also: ["Using the Message Logs"](#) on page 1-11

Invalid Input Errors

In the Web services API, these errors are `InvalidInputFault` SOAP faults. They are general input errors.

EQA-10000: An object type must be specified for operation *operationName*.

Cause: The operation did not contain an object type.

Action: Specify an object type.

EQA-10001: The operation *operationName* is not supported for object type *objectType*.

Cause: An invalid operation was specified for the object type.

Action: Use a valid operation, such as `export` instead of `exportAll`.

EQA-10002: The value *inputValue* is not a supported object type.

Cause: The operation specified an object type that is not supported by the Administration API.

Action: Verify that you entered the object type correctly.

EQA-10003: An object key must be specified for operation *operationName*.

Cause: The operation did not include an object key.

Action: Include the object key for a creatable object.

EQA-10004: Invalid object key *objectKey*. Specify *keyName1*, *keyName2*

Cause: An invalid or incomplete object key was specified.

Action: Specify the required key names and values.

EQA-10006: Specify at least one object key for operation *operationName*.

Cause: The operation required a list of object keys representing the objects to process, but no object keys were specified.

Action: Specify at least one object key.

EQA-10007: The input XML must be specified for operation *operationName*.

Cause: The operation did not include the input XML.

Action: Specify the input XML.

EQA-10008: An unexpected error occurred while reading the input XML.

Cause: An unexpected error occurred while reading the XML.

Action: Inspect the underlying exception message, the log files, or both.

EQA-10009: The input XML is not a valid XML document.

Cause: The input XML was not a well-formed XML document.

Action: Inspect the underlying exception message and correct the syntax errors in the input XML.

EQA-10010: The input XML does not conform to the XML schema.

Cause: The input XML did not conform to the XML schema.

Action: Inspect the underlying exception message and correct the input XML to conform to the schema.

EQA-10011: The object with type *universalType* was not found in the input XML.

Cause: The input XML did not contain the specified object type.

Action: Ensure that the input XML contains an object description for the universal type specified in the operation.

EQA-10012: The object with key *objectKey* and type *creatableType* was not found in the input XML.

Cause: The input XML did not contain the specified creatable object.

Action: Ensure that the input XML contains an object description for the creatable object specified in the operation.

EQA-10015: The duplicate method *duplicateMethod* is not supported for object type *objectType*.

Cause: A duplicate method was specified that is not supported for this object type.

Action: Specify a different duplicate method, or omit the duplicate method to use the default.

EQA-10016: The value *inputValue* is not a supported duplicate method.

Cause: A duplicate method was specified that is not supported for any object type.

Action: Specify a supported duplicate method, or omit the duplicate method to use the default.

EQA-10017: The update method must be specified for operation *operationName*.

Cause: The operation did not contain an update method.

Action: Specify an update method for the operation.

EQA-10018: The update method *method* is not supported for object type *objectType*.

Cause: An update method was specified that is not supported for this object type.

Action: Specify a supported update method.

EQA-10019: The value *inputValue* is not a supported update method.

Cause: A value was specified that is not a valid update method for any object type.

Action: Specify a supported update method.

EQA-10021: The "not found method" *notFoundMethod* is not supported for object type *objectType*.

Cause: A "not found method" was specified that is not supported for this object type.

Action: Specify a supported "not found method" for the object type.

EQA-10022: The value *inputValue* is not a supported "not found method."

Cause: A value was specified that is not a valid "not found method" for any object type.

Action: Specify a supported "not found method."

EQA-10024: The property *propertyName* for object type *objectType* must be specified.

Cause: The object definition did not contain a required property.

Action: Specify the property.

EQA-10025: The property *propertyName* for object type *objectType* cannot be empty.

Cause: The object definition omitted a value for a required property.

Action: Enter a value for the property in the object definition.

EQA-10026: The property *propertyName* for object type *objectType* is malformed.

Cause: A property value in the object description was not well-formed according to the expected format. For example, if the property represents a URL, the property value must be a well-formed URL.

Action: Specify a well-formed property value. Inspect the underlying exception message, the log files, or both for more information.

EQA-10027: The property *propertyName* for object type *objectType* must be between *value1* and *value2*.

Cause: The specified value of a property was outside the valid range.

Action: Specify a value within the valid range.

EQA-10028: The value *inputValue* is not supported for property *propertyName* and object type *objectType*.

Cause: The specified value of a property was not a supported value.

Action: Specify a valid value.

EQA-10029: The property *propertyName* for object type *objectType* must be an absolute path.

Cause: A file path was specified that is not an absolute path. The specified value cannot be a relative path.

Action: Specify an absolute path.

EQA-10030: The property *propertyName* for object type *objectType* cannot contain: *inputValue*

Cause: The property prohibited the use of the input value.

Action: Specify a valid value.

EQA-10031: The property *propertyName* for object type *objectType* can only contain ASCII characters.

Cause: The object definition contained invalid characters, such as multibyte characters, in a property value.

Action: Specify an ASCII value.

EQA-10032: The resource specified in property *propertyName* for object type *objectType* was not found.

Cause: A property value identified an external resource, such as a file, that was not found.

Action: Ensure that the specified resource exists.

EQA-10033: The property *propertyName* for object type *objectType* must not be empty for operation *operationName*.

Cause: The object definition omitted the value of a property that is required for this operation and object type.

Action: Specify a value for the property.

EQA-10034: The value *inputValue1* for property *propertyName1* of object type *objectType* is not valid for value *inputValue2* of property *propertyName2*.

Cause: The object definition specified a value for *propertyName1* that is invalid when *propertyName2* has a value of *inputValue2*.

Action: Specify a valid combination of values for the two properties.

EQA-10035: The properties *propertyName1* and *propertyName2* for object type *objectType* must have different values.

Cause: The object definition specified the same value for the two properties.

Action: Change one of the property values.

EQA-10037: The value *inputValue* for property *propertyName* of object type *objectType* is a reserved value.

Cause: A value was specified for the property and object type that is reserved for internal use by Oracle SES.

Action: Specify a valid property value.

EQA-10038: The property *propertyName* for object type *objectType* must start with *value*.

Cause: A property name began with one or more invalid characters.

Action: Correct the name so that it begins with the specified value.

EQA-10039: The property *propertyName* for object type *objectType* must end with *value*.

Cause: A property name ended with one or more invalid characters.

Action: Correct the name so that it ends with the specified value.

EQA-10040: The property *propertyName* for object type *objectType* must not end with *inputValue*.

Cause: A property name ended with one or more invalid characters.

Action: Correct the name so that it does not end with the specified value.

EQA-10041: The property *propertyName* for object type *objectType* must be specified when property *propertyName* is specified.

Cause: The object definition did not contain a required property.

Action: Specify both properties or neither of them.

EQA-10042: The value *inputValue* for property *propertyName* of object type *objectType* is already in use.

Cause: A property value was already in use by another object of the same type, and only one object of this type can have this value.

Action: Specify a different value for the property, or remove the value from the other object that is using it.

EQA-10043: The value *inputValue* for property *propertyName* of object type *objectType* contains an invalid protocol.

Cause: A property value contained an invalid protocol. For example, the protocol for a Web source starting URL must be `http` or `https`.

Action: Specify a valid protocol for the property value.

EQA-10044: The value *inputValue* for property *propertyName* of object type *objectType* contains an invalid separator.

Cause: A file URL prefix contained an invalid character for the separator.

Action: Use a slash (/) instead of a backslash (\) as the separator in a URL.

EQA-10045: At least one of the properties *propertyName1* or *propertyName2* for object type *objectType* must have the value *inputValue*.

Cause: The object definition did not contain a property with the required value.

Action: Change one of the specified properties to the required value.

EQA-10046: The value *inputValue* of property *propertyName* for object type *objectType* is not supported. The supported values are: *value1, value2 . . .*}

Cause: A property contained an unsupported value.

Action: Change the property value to a supported values.

EQA-10047: The value of property *propertyName1* for object type *objectType* must be greater than or equal to the value of property *propertyName2*.

Cause: A property value was too small.

Action: Increase the property value so that it is as least as large as the other property value.

EQA-10048: The property *propertyName* for object type *objectType* must start with an alphabetic character and may only contain alphanumeric characters and `_`, `$`, and `#`.

Cause: A property contained one or more invalid characters.

Action: Use only valid characters for the property value

EQA-10049: The property *propertyName* for object type *objectType* must be in the format of *format*.

Cause: A property value had an invalid format.

Action: Specify the value in the specified format.

EQA-10050: The property *propertyName1* for object type *objectType* can only contain one value when property *propertyName2* has value *inputValue*.

Cause: A property contained multiple values, whereas only one value is allowed.

Action: Specify only one value for the property.

EQA-10051: Key patterns are not supported for operation *operationName* and object type *objectType*.

Cause: An invalid option was specified for this operation and object type. The operation can support key patterns for some object types, but not for the specified object type.

Action: Do not specify a key pattern for this object type.

EQA-10052: The values of properties *propertyName1* and *propertyName2* for object type *objectType* must be the same.

Cause: The object description contained two properties with different values, when they must have the same value.

Action: Change one of the property values to duplicate the other one.

EQA-10053: The value *inputValue* of property *propertyName* for object type *objectType* must be writable.

Cause: The property value was not a writable location. For example, the property may represent a file system path, such as the crawler log file directory for a `crawlerSettings` object.

Action: Ensure that you are connected to the Administration API as a user with write privileges to the specified location.

EQA-10054: The value for property *propertyName1* of object type *objectType* is not valid for value *inputValue* of property *propertyName2*.

Cause: The value for *propertyName1* is invalid when *propertyName2* has a value of *inputValue*.

Action: Specify a valid combination of values for the two properties.

EQA-10055: The value for property *propertyName* of object type *objectType* is not a directory path.

Cause: A property contained an invalid directory path.

Action: Change the property value to an existing directory path.

EQA-10056: The value of property *propertyName* for object type *objectType* cannot be changed.

Cause: A property contained a new value, but it cannot be changed after the object is created.

Action: Specify the current value of the property or omit the property from the object description.

EQA-10057: The value *inputValue* of property *propertyName* for object type *objectType* does not match the expected value *expectedValue*.

Cause: A property contained a value that did not match the expected value. This error may occur when the property must have a specific value. For example, a property containing the version number of an identity plug-in must match the version of the Java plug-in implementation.

Action: Specify the expected value for the property.

EQA-10058: The value of property *propertyName* for object type *objectType* does not match the expected value.

Cause: The property contained a list of values that did not match the expected list. For example, a property containing the security attributes for a source must match the plug-in for that source type.

Action: Correct the property value.

EQA-10200: The class *inputValue* does not implement the plug-in manager interface *managerInterface*.

Cause: A Java class was specified that did not implement the plug-in manager interface.

Action: Specify a Java class that implements the plug-in manager interface.

EQA-10201: The class *inputValue* implements the plug-in interface instead of the plug-in manager interface. Use the manager class name.

Cause: A Java class was specified that implements the plug-in interface instead of the plug-in manager interface.

Action: Specify a Java class that implements the plug-in manager interface.

EQA-10202: The plug-in parameters were rejected by the plug-in manager. See the log file for more information.

Cause: A plug-in manager validated the plug-in parameters, and it rejected the input values.

Action: Inspect the log file for more information.

EQA-10203: Unable to instantiate the plug-in manager class *inputValue*. Ensure that the class contains an empty constructor.

Cause: A problem occurred while loading the plug-in manager class.

Action: Ensure the class contains an empty constructor. Inspect the log file for more information.

EQA-10204: The input XML must be specified for operation *operationName* and type *objectType*.

Cause: The operation did not include the input XML, which it requires for this object type. For example, the `activate` operation requires input XML for identity plug-ins, but not for clustering.

Action: Include the input XML in the operation.

EQA-10206: The key name *keyName* is not valid for object type *objectType*.

Cause: A key name was specified that is not valid for this object type.

Action: Specify a valid key name for the object type.

EQA-10207: A value must be specified for key name *keyName*.

Cause: A key name was specified without a key value.

Action: Specify a value for the key name.

EQA-10208: The manager class *inputValue* does not provide plug-ins that implement the interface *interfaceName*.

Cause: The specified manager class did not provide plug-in instances that implement the correct interface.

Action: Correct the manager class to return plug-in instances that implement the specified interface.

- EQA-10209: The security attributes were rejected by the plug-in manager *inputValue*. See the log file for more information.**
Cause: The plug-in manager rejected the security attributes.
Action: Inspect the log file for more information.
- EQA-10210: The object with key *objectKey* and type *creatableType* occurs more than once in the input to operation *operationName*.**
Cause: An object was specified multiple times for the operation, but only one version of an object can be used as input to an operation.
Action: Remove all but one reference to the object.
- EQA-10211: The object key *objectKey* occurs more than once in the key list for operation *operationName*.**
Cause: An object key was specified multiple times in the key list for the operation.
Action: Remove all but one reference to the object.
- EQA-10212: The value of property *propertyName* in the object with key *objectKey* and type *creatableType* could not be encrypted. See the log file for more information.**
Cause: A value was specified that could not be encrypted.
Action: Inspect the log file for more information.
- EQA-10213: The value of property *propertyName* in the object with type *universalType* could not be encrypted. See the log file for more information.**
Cause: A value was specified that could not be encrypted.
Action: Inspect the log file for more information.
- EQA-10214: The value of property *propertyName* in the object with key *objectKey* and type *creatableType* could not be decrypted. Ensure that the encryption key is correct and that the property contains an encrypted value.**
Cause: A property value could not be decrypted.
Action: Ensure that the property contains a value encrypted by Oracle SES, and the decryption key duplicates the original encryption key.
- EQA-10215: The value of property *propertyName* in the object with type *universalType* could not be decrypted. Ensure that the encryption key is correct and that the property contains an encrypted value.**
Cause: The specified value could not be decrypted.
Action: Ensure that the property contains a value encrypted by Oracle SES, and the decryption key duplicates the original encryption key.
- EQA-10216: An encryption key must be specified for operation *operationName*.**
Cause: The operation did not include an encryption key.
Action: Include an encryption key in the operation.
- EQA-10217: The encryption key must be at least *value* characters.**
Cause: The encryption key was too short.
Action: Enter an encryption key with at least the minimum number of characters.
- EQA-10218: The encryption key must contain both letters and numbers.**
Cause: The encryption key contained letters or numbers, but not both as required.
Action: Enter an encryption key that contains both letters and numbers.

- EQA-10219: The object type for operation *operationName* must be a stateful type.**
Cause: The operation specified an object type that does not have state properties.
Action: Specify a valid operation for the object type.
- EQA-10220: The value *inputValue* is not a supported state property name for object type *objectType*.**
Cause: A state property was specified that is invalid for the object type.
Action: Specify a supported state property for the object type.
- EQA-10221: The value *inputValue* is not a supported operation control name.**
Cause: The specified control name was invalid.
Action: Specify a valid setting for the operation control.
- EQA-10222: The operation control *inputValue* is not supported for operation *operationName*.**
Cause: A control name was specified that is invalid for this operation.
Action: Use either valid operation control or none.
- EQA-10223: The value *inputValue* is not supported for operation control *controlName*. The supported values are: *value1*, *value2***
Cause: The value of the operation control was invalid.
Action: Specify one of the listed values.
- EQA-10224: An object key cannot be specified for the universal object type *universalType*.**
Cause: An object key was specified for a universal object type.
Action: Omit the object key from the operation, or specify the correct object type for the key.
- EQA-10225: The attachment with resource name *resourceName* for the object with key *objectKey* and type *creatableType* was not found.**
Cause: The specified attachment was not found.
Action: Ensure that the resource name is correct.
- EQA-10226: The attachment with resource name *resourceName* for the object with type *universalType* was not found.**
Cause: The specified attachment was not found.
Action: Ensure that the resource name is correct.
- EQA-10227: The attachment with resource name *resourceName* occurs more than once in the attachment list for operation *operationName*.**
Cause: The attachment list referenced a resource name multiple times.
Action: Remove the duplicate entry from the attachment list.
- EQA-10228: A resource name must be specified for all attachments in operation *operationName*.**
Cause: An attachment was specified without a resource name.
Action: Provide a resource name for each attachment.
- EQA-10229: The value *inputValue* is not a valid user for the active identity plug-in.**

Cause: The specified value is not a valid user according to the active identity plug-in. For example, a property may have to be an LDAP user name corresponding to the identity plug-in, but the value specified was not.

Action: Specify the correct user name and password for the active identity plug-in.

EQA-10230: The encryption key can only contain ASCII characters.

Cause: An encryption key contained invalid characters, such as double-byte characters.

Action: Use only ASCII characters in the encryption key.

Creatable Administration Object Errors

In the Web services API, these errors are `CreatableAdminObjectFault` SOAP faults. They apply only to creatable object types.

EQA-11000: The object with key *objectKey* and type *creatableType* was not found.

Cause: An object with the given key and type did not exist.

Action: Ensure that the object key and type are correct and the object actually exists.

EQA-11001: The object with key *objectKey* and type *creatableType* already exists.

Cause: An object with the given key and type was already defined.

Action: Perform an `update` operation instead of `create` to revise the definition of an existing object. Or, in a `createAll` operation, specify `overwrite` or `ignore` for the duplicate method.

EQA-11002: The operation *operationName* is not supported for the object with key *objectKey* and type *creatableType*.

Cause: The specified operation cannot be performed on the object.

Action: If possible, use a related, supported operation for the object, such as `delete` instead of `deleteList`. Otherwise, none. The operation is not supported.

EQA-11003: The maximum number of objects allowed with type *creatableType* and value *keyValue* for key *keyName* is *maxObjects*.

Cause: The object was not created, because the maximum number of objects with the specified name and value already existed. For example, an `altWord` object can have a maximum of four alternate words with the same keyword.

Action: Delete one or more existing objects before trying to create new ones with the specified name and value.

EQA-11004: The object with key *objectKey* and type *creatableType* is not an instance of type *creatableType* that is currently supported.

Cause: The specified key and object type were valid, but the object cannot be managed through the Administration API. For example, the Administration API can manage sources but cannot manage all source types.

Action: Use a method of managing the object that is currently supported, such as the Administration GUI.

Invalid State Errors

In the Web services API, these errors are `InvalidStateFault` SOAP faults. They are caused by the current state of the object, such as failing to delete a schedule because it is currently executing.

EQA-13000: Operation *operationName* cannot be performed on an object with type *objectType* in state *objectStatus*.

Cause: An operation was specified that could not be performed on the object while it was in its current state.

Action: Correct the state of the object before proceeding.

EQA-13001: The object with key *objectKey* and type *creatableType* is already active.

Cause: The activate operation was attempted on a creatable object that was already active.

Action: None. The object is already active.

EQA-13002: The object with type *universalType* is already active.

Cause: The activate operation was attempted on a universal object that was already active.

Action: None. The object is already active.

EQA-13003: The object with key *objectKey* and type *creatableType* is already inactive.

Cause: The deactivate operation was attempted on a creatable object that was already inactive.

Action: None. The object is already inactive.

EQA-13004: The object with type *universalType* is already inactive.

Cause: The deactivate operation was attempted on a universal object that was already inactive.

Action: None. The object is already inactive.

EQA-13005: The object with key *objectKey* and type *creatableType* cannot be activated because another object with the same type is already active.

Cause: The activate operation was attempted on a creatable object, but another object of the same type was already active. Only one object of this type can be active at a time, such as an identity plug-in.

Action: Deactivate the other object of this type, then try again.

EQA-13006: The object with key *objectKey* and type *creatableType* is already starting.

Cause: The start operation was attempted on a creatable object that was already starting.

Action: To restart the object, first stop it or wait for it to stop.

EQA-13007: The object with type *universalType* is already starting.

Cause: The start operation was attempted on a universal object that was already starting.

Action: To restart the object, first stop it or wait for it to stop.

EQA-13008: The object with key *objectKey* and type *creatableType* is already stopping.

Cause: The stop operation was attempted on a creatable object that was already stopping.

Action: None. Wait for it to stop.

EQA-13009: The object with type *universalType* is already stopping.

Cause: The stop operation was attempted on a universal object that was already stopping.

Action: None. Wait for it to stop.

Administration API Run-Time Errors

In the Web services API, these errors are `AdminRuntimeFault` SOAP faults. They are unexpected or result from conditions encountered at run-time, such as an invalid user name or an unavailable database. In the Java client, these are unchecked exceptions.

EQA-15000: An unexpected error occurred during operation *operationName*.

Cause: An unexpected error occurred during an API operation.

Action: Inspect the underlying error message, the log files, or both.

EQA-15001: A connection to the database could not be established.

Cause: An attempt to connect to the database failed.

Action: Inspect the underlying exception in the log file.

EQA-15002: A connection to the Web Service URL *wsURL* could not be established.

Cause: An attempt to connect to the Web service URL failed.

Action: Verify that the Web service endpoint is running by opening `http://host:port/search/api/admin/AdminService?WSDL` in a browser. If it is running and the problem remains, then inspect the underlying exception in the log file.

EQA-15003: Invalid login credentials. Check your user name and password and try again.

Cause: An invalid user name, password, or both were provided.

Action: Enter the correct user name and password, and try again.

EQA-15004: The operation *operationName* can only be performed by logged in users.

Cause: A Web services operation was attempted without previously logging in for stateful mode or providing credentials for stateless mode.

Action: For stateful mode, call `login` first. For stateless mode, provide the `Credentials` argument for the operation.

EQA-15005: An unexpected error occurred while marshalling the XML document.

Cause: An error occurred while constructing the output XML.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15006: An unexpected error occurred while unmarshalling the XML document.

Cause: An error occurred while processing the input XML.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15007: The XML schema *xsdName* was not found.

Cause: A required XML schema was not found.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15008: An unexpected error occurred while reading the XML schema.

Cause: An error occurred while reading a required XML schema.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15009: An unexpected error occurred while reading from the database.

Cause: An error occurred while reading from the database.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15010: The plug-in manager raised an error while validating the plug-in parameters.

Cause: A plug-in manager class raised an error during validation of the plug-in parameters in the input XML.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15011: The plug-in manager raised an unexpected error.

Cause: A plug-in manager raised an unexpected error.

Action: Inspect the underlying exception message, the log files, or both.

EQA-15014: Operation *operationName* cannot proceed because the remote resource located at *resourceURI* cannot be contacted.

Cause: An operation was unable to contact a remote resource (such as a federated source), which was needed to continue.

Action: Ensure that the remote resource is available and try again.

EQA-15015: The external tool *externalToolName* raised an error.

Cause: The API invoked an external executable tool, which raised an error.

Action: Inspect the underlying error message, the log files, or both.

Dependent Object Errors

In the Web services API, these errors are `DependentObjectFault` SOAP faults. They are related to dependent objects of the object being processed. For example, an operation on a schedule might fail because of the current state of a source.

EQA-16000: The dependent object *dependentObjectName* referenced in object with type *universalType* was not found.

Cause: A universal administrative object referenced another object that did not exist, such as a `resultList` with rendering attributes that were not created previously.

Action: Create the dependent object first.

EQA-16001: The dependent object *dependentObjectName* referenced in object with key *objectKey* and type *creatableType* was not found.

Cause: A creatable administrative object referenced another object that did not exist, such as a `schedule` with an assigned `source` that was not created previously.

Action: Create the dependent object first.

EQA-16002: A duplicate dependent object *dependentObjectName* was found in object with type *universalType*.

Cause: A universal administrative object referenced a list of dependent objects that contained duplicates, such as a `resultList` that listed the same rendering attributes twice.

Action: Remove the duplicates in the dependent object list.

EQA-16003: A duplicate dependent object *dependentObjectName* was found in object with key *objectKey* and type *creatableType*.

Cause: A creatable administrative object referenced a list of dependent objects that contained duplicates, such as a `schedule` that listed the same assigned source twice.

Action: Remove the duplicates in the dependent object list.

EQA-16004: The dependent object *dependentObjectName* referenced in object with type *universalType* is in use.

Cause: A universal administrative object referenced a dependent object that was already being referenced. Only one object can reference the dependent object.

Action: Remove the reference to the dependent object from the other administrative object. For example, remove the rendering attribute from the other `resultList`.

EQA-16005: The dependent object *dependentObjectName* referenced in object with key *objectKey* and type *creatableType* is in use.

Cause: A creatable administrative object referenced a dependent object that was already being referenced. For example, a `schedule` cannot be created with an assigned source that is already assigned to another `schedule`. Only one object can reference the dependent object.

Action: Remove the reference to the dependent object from the other administrative object. For example, remove the assigned source from the other `schedule`.

EQA-16006: The object with type *universalType* is in use by dependent object *dependentObjectName*.

Cause: The universal administrative object was in use by a dependent object.

Action: Eliminate the dependency before attempting this operation again.

EQA-16007: The object with key *objectKey* and type *creatableType* is in use by dependent object *dependentObjectName*.

Cause: The creatable administrative object was in use by the dependent object. For example, a search attribute cannot be deleted while a source attribute mapping is using it.

Action: Eliminate the dependency. For example, first remove the source with the attribute mapping or remove the attribute mapping from the source, then delete the search attribute.

EQA-16008: The dependent object *dependentObjectName* referenced in object with type *universalType* was found but is not valid for this object.

Cause: A universal administrative object referenced a dependent object that was invalid for this reference, such as a Date attribute where only String attributes are valid.

Action: Reference a dependent object with valid characteristics. You may modify the characteristics of the specified dependent object or reference a different dependent object.

EQA-16009: The dependent object *dependentObjectName* referenced in object with key *objectKey* and type *creatableType* was found but is not valid for this object.

Cause: A creatable administrative object referenced a dependent object that was invalid for this reference, such as a Date attribute where only String attributes are valid.

Action: Reference a dependent object with valid characteristics. You may modify the characteristics of the specified dependent object or reference a different dependent object.

EQA-16010: Operation *operationName* cannot be performed on object with type *universalType* because of the state of a dependent object with type *dependentObjectType*.

Cause: The universal administrative object depends on an object that was in an invalid state for the operation.

Action: Correct the state of the dependent object.

EQA-16011: Operation *operationName* cannot be performed on object with key *objectKey* and type *creatableType* because of the state of a dependent object with type *dependentObjectType*.

Cause: The creatable administrative object depends on an object that was in an invalid state for the operation. For example, a source cannot be deleted while assigned to a schedule that is executing.

Action: Correct the state of the dependent object.

EQA-16012: The property *propertyName* for object with type *universalType* must contain the following dependent objects: *dependentObjectName1*, *dependentObjectName2*

Cause: A property in the universal administrative object did not contain the required dependent objects.

Action: Modify the property to contain the listed dependent objects.

EQA-16013: The property *propertyName* for object with key *objectKey* and type *creatableType* must contain the following dependent objects: *dependentObjectName1*, *dependentNameObject2*

Cause: A property in the creatable administrative object did not contain the required dependent objects, such as the mandatory attribute names for a federated source.

Action: Modify the property to contain the listed dependent objects.

EQA-16014: The property *propertyName* for object with type *universalType* requires a dependent object with type *dependentObjectType* to be active.

Cause: A dependent object was not active, as required by the universal administrative object.

Action: Activate an appropriate dependent object with the specified type.

EQA-16015: The property *propertyName* for object with key *objectKey* and type *creatableType* requires a dependent object with type *dependentObjectType* to be active.

Cause: A dependent object was not active, as required by the creatable administrative object. For example, an active identity plug-in is needed for using a source-level ACL policy in a source.

Action: Activate an appropriate dependent object with the specified type.

EQA-16016: The dependent object *dependentObjectName* referenced in object with type *universalType* was not found as specified. The specified value *specifiedValue* for property *propertyName* differs from the actual value *actualValue*.

Cause: A universal administrative object referenced the properties of a dependent object, but the specified values did not match the actual values. For example, the data type of a search attribute might be specified incorrectly.

Action: Correct the specification of the dependent object.

EQA-16017: The dependent object *dependentObjectName* referenced in object with key *objectKey* and type *objectType* was not found as specified. The specified value *specifiedValue* for property *propertyName* differs from the actual value *actualValue*.

Cause: A creatable administrative object referenced the properties of a dependent object, but the specified values did not match the actual values. For example, the data type of a search attribute might be specified incorrectly.

Action: Correct the specification of the dependent object.

EQA-16019: Usage of the value *propertyValue* for property *propertyName* requires that the parent object with key *objectKey* and type *creatableType* has not been crawled.

Cause: The source description contained a property value that cannot be used after the source has been crawled.

Action: Either delete and re-create the source with this property value, or specify a different value for the property.

EQA-16020: The operation *operationName* for object with type *universalType* requires a dependent object with type *dependentObjectType* to be active.

Cause: A dependent object was not active, as required for the operation on the specified universal administrative object.

Action: Activate the specified dependent object.

EQA-16021: The operation *operationName* for object with key *objectKey* and type *creatableType* requires a dependent object with type *dependentObjectType* to be active.

Cause: A dependent object was not active, as required for the operation on the specified creatable administrative object. For example, an identity plug-in must be active when exporting a source that uses source-level ACL and access control entries.

Action: Activate the specified dependent object.

A

about operator, 2-106
ACL policies, 2-83
activate command, 3-1
activate service request, 4-3
address (Web service), 1-5
administration API run-time errors, B-12
administration object errors, B-10
administrator log-in, 1-4
alert
 clearing, 2-12, 2-24
 deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 object description, 2-23
 returning object keys, 4-33
 updating, 3-96, 3-117, 4-52, 4-56
alternate words, *See* altWord
altWord
 creating, 3-12, 3-25, 4-6, 4-9
 deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 object description, 2-26
 returning object keys, 3-57, 4-33
 updating, 3-97, 3-118, 4-52, 4-56
and operator, 2-106
API run-time errors, B-12
API version, 3-59, 4-35
applications, searching, 2-68
ASP.NET, 0-xiii, 1-5
asset files, 2-14, 2-66
attributes
 custom search, 2-63
 in search results, 2-57
authentication
 plug-ins, 2-38
 self-service, 2-83
 time-out, 2-54
authorization
 plug-ins, 2-79
 time-out, 2-54

B

boundary rules, 2-54, 2-84
BT (broader term), 2-109

Bubbling library, 2-15
bye, *See* quit command

C

cache files, 2-103
cache settings, 2-36
cascading style sheets, 2-14, 2-58, 2-66
character sets, 2-5
cluster nodes, *See* clustering
cluster trees, *See* clusterTree
clustering
 activating, 3-4, 4-3
 deactivating, 3-36, 4-13
 exporting, 3-51, 4-24
 object description, 2-28
 pipeline, 2-36
 returning current state, 3-68, 4-40
 updating, 3-98, 4-52
clusterTree
 activating, 3-5, 4-3
 creating, 3-13, 3-26, 4-6, 4-9
 deactivating, 3-37, 4-13
 deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 object description, 2-30
 returning current state, 3-60, 3-69, 3-84, 4-37, 4-40, 4-43
 returning object keys, 3-57, 4-33
 updating, 3-99, 3-119, 4-52, 4-56
collaboration repositories, searching, 2-68
configuration settings
 query results, 2-51
 search application, 2-15
Config.xsd, 1-10
CONNECTION option, 1-3, 1-4
CONNECTION_LIST option, 1-4
content management repositories, searching, 2-68
cookies, 2-87
count method, 2-53
crawler
 character sets, 2-5
 documents formats, 2-2
 languages, 2-4
 log configuration, 2-36
 log files, 2-103

- scheduling, 2-59
- crawler configuration, 2-74, 2-80, 2-85
 - See also* crawlerSettings
- crawler settings
 - See* crawlerSettings
- crawlerSettings
 - exporting, 3-51, 4-24
 - object description, 2-34
 - updating, 3-100, 4-52
- creatable types
 - listed, 1-8
 - retrieving object keys, 3-57
- create command, 3-1
- create service request, 4-6
- createAll command, 3-1
- createAll service request, 4-9
- credentials, 4-46
- custom attributes, 2-63
- customizations, user interface, 2-66

D

- dash, using in commands, 3-3
- data sources, 2-68
- database storage, 2-102
- database tables, searching, 2-68
- deactivate command, 3-2
- deactivate service request, 4-13
- delete command, 3-2, 3-44
- delete service request, 4-15
- deleteAll command, 3-2, 3-45
- deleteAll service request, 4-17
- deleteList command, 3-2, 3-47
- deleteList service request, 4-20
- dependent object errors, B-13
- describe, *See* export
- dictionary, 2-53
- directives, 2-14
- disk quotas, 2-12
 - alert, 2-23
 - spaceCalculator, 2-99
 - storageArea, 2-102
- display URLs, 2-52
- do you mean message, 2-27
- document cache settings, 2-36
- document formats, 2-2
- document service pipeline, 2-36
- documents, searching, 2-68

E

- e-mail display URL, 2-52
- encryption, 2-8
- endpoint, 1-5
- English dictionary, 2-53
- error messages, B-1
- errors
 - administration objects, B-10
 - API run-time, B-12
 - dependent objects, B-13

- invalid input, B-1
- invalid state, B-11
- exit, *See* quit command
- export command, 3-49, 3-51
- export service request, 4-24
- exportAll command, 3-53
- exportAll service request, 4-27
- exporting object descriptions, 3-49 to 3-56
- exportList command, 3-55
- exportList service request, 4-30

F

- faults
 - See* errors
- federated search, trusted entities, 2-49
- federated sources, 2-54, 2-69 to 2-71
- federation endpoints, 2-49
- file display URL, 2-52, 2-73
- file size, 2-102
- file sources, 2-72
- files, searching, 2-68
- filter rules, 2-70
- forms, 2-88
- fragmentation, index, 2-42, 2-44
- Freemarker templates, 2-14

G

- getAllObjectKeys command, 3-57
- getAllObjectKeys service request, 4-33
- getAllStates command, 3-2
- getAllStates service request, 4-37
- getAPIVersion command, 3-59
- getAPIVersion service request, 4-35
- getState command, 3-2
- getState service request, 4-40
- getStateList command, 3-2
- getStateList service request, 4-43
- globalization, 2-3
- graphics, 2-14

H

- hash partitioning, 2-47
- help command, 1-3
- hit count method, 2-53
- HTML forms, 2-88
- hyphen, using in commands, 3-3

I

- identity plug-ins, *See* identityPlugin
- identityPlugin
 - activating, 3-6, 4-3
 - creating, 3-14, 3-27, 4-6, 4-9
 - deactivating, 3-38, 4-13
 - deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 - exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 - object description, 2-38
 - returning current state, 3-62, 3-71, 3-86, 4-37, 4-40,

- 4-43
 - returning object keys, 3-57, 4-33
- images, 2-14
- implementation class, 1-5
- inc_footer.ftl, 2-19, 2-20, 2-21
- inc_header.ftl, 2-19
- inc_logo_querybox.ftl, 2-19, 2-20
- index
 - exporting, 3-51, 4-24
 - object description, 2-42
 - returning current state, 3-73, 4-40
 - updating, 3-101, 4-52
- index fragmentation, 2-42
- index optimizer, *See* indexOptimizer
- index refresh, 2-59
- indexOptimizer
 - activating, 3-7, 4-3
 - deactivating, 3-39, 4-13
 - exporting, 3-51, 4-24
 - object description, 2-44
 - returning current state, 3-74, 4-40
 - starting, 3-91, 4-48
 - updating, 3-102, 4-52
- interactive sessions, 1-2
- invalid input errors, B-1
- invalid state errors, B-11

J

- Java client (Web service), 1-5
- Java example, A-1
- Java Runtime Environment (JRE), A-5
- Javascript libraries, 2-15
- JRockit, A-5

K

- key names, retrieving, 3-57, 4-33

L

- language codes
 - crawler, 2-4
 - product, 2-3
 - query, 2-7
- language detection, 2-35, 2-75
- language support, 2-3, 2-6
- language, default, 2-35, 2-75
- links, depth limit, 2-35
- list, *See* export
- lists of values (LOVs), 2-63
- localization, 2-3
- location, sources types, 2-95
- log-in proxies, *See* proxyLogin
- log-in service requests, 4-46
- logo, custom, 2-17
- logout service request, 4-47
- logs
 - crawler, 2-36, 2-103
 - message, 1-11
 - operations, 1-11

- querying statistics, 2-53
- search statistics, 2-53

M

- macros, 2-14
- mailing list display URL, 2-52
- message log, 1-11
- metadata cluster tree, 2-31
- metadata repository, 2-42
- MIME types, 2-2

N

- near operator, 2-106
- nested link limit, 2-35
- NT (narrower term), 2-109

O

- object errors, B-10, B-13
- object keys, retrieving, 3-57
- objects
 - creatable, 1-8
 - universal, 1-7
- operations log, 1-11
- operations, listed by function, 1-12
- operators, in search queries, 2-106
- optimization, index, 2-42, 2-44
- or operator, 2-106
- Oracle SES API version, 3-59, 4-35
- OracleAS Single Sign-On, 2-89

P

- parallel query, 2-10
- partition configuration
 - See* partitionConfig
- partitionConfig
 - exporting, 3-51
 - object description, 2-47
 - updating, 3-103
- partitioning, 2-10
- partitions, 2-103
- password encryption, 2-8
- PASSWORD option, 1-4
- passwords, 4-46
- percent signs, using in commands, 3-3
- phrase operator, 2-106
- port name, 1-5
- predefined directives, 2-14
- print, *See* export
- processes
 - for searching, 2-54
 - resuming after alert, 2-107
- proxies, *See* proxyLogin
- proxyLogin
 - creating, 3-15, 3-28, 4-6, 4-9
 - deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 - exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 - object description, 2-49

returning object keys, 3-57, 4-33
updating, 3-104, 3-120, 4-52, 4-56

Q

query configuration, *See* queryConfig

query expansion

alternate words, 2-26
suggested links, 2-105
thesaurus, 2-108

query interface, style sheets, 2-58

query languages, 2-7

query log, 2-53

query routing, 2-70

query statistics, 2-53

queryConfig

exporting, 3-51, 4-24
object description, 2-51
updating, 3-105, 4-52

QueryPlan.xml configuration settings, 2-15

quit command, 1-2

quotas, disk

See disk quotas

R

relevancy boosting, 2-52

result list settings, *See* resultList

resultList

activating, 3-8, 4-3
deactivating, 3-40, 4-13
exporting, 3-51, 4-24
object description, 2-57
returning current state, 3-76, 4-40
updating, 3-106, 4-52

results, *See* search results

results.ftl, 2-19

robots exclusion, 2-86

RSS feed, 2-17

run-time errors, B-12

S

schedule

activating, 3-9, 4-3
creating, 3-16, 3-29, 4-6, 4-9
deactivating, 3-41, 4-13
deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
object description, 2-59
returning current state, 3-64, 3-77, 3-88, 4-37, 4-40, 4-43
returning object keys, 3-57, 4-33
starting, 3-92, 4-48
stopping, 3-95, 4-50
updating, 3-107, 3-121, 4-52, 4-56

search application, configuration settings, 2-15

search attributes, *See* searchAttr

search interface customizations, 2-13 to 2-22

search languages, 2-7

search results

limiting by cluster tree, 2-30

limiting by source group, 2-93

search statistics, 2-53

search statistics log, 2-53

searchadmin

alphabetical command list, 3-1

command syntax, 1-10

individual commands, 1-2

interactive sessions, 1-2

searchAttr

creating, 3-17, 3-30, 4-6, 4-9

deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20

exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30

object description, 2-63

returning object keys, 3-57, 4-33

updating, 3-108, 3-122, 4-53, 4-56

secure search, 2-38

self-service authentication, 2-83

service name, 1-5

SES API version, 3-59, 4-35

seslib.ftl, 2-15

session mode

command-line interface, 1-2

sessions, opening stateful, 4-46

single job mode

command-line interface, 1-2

single sign-on, 2-89

skin bundles, *See* skinBundle, 2-66

skin bundles, specifying in URLs, 2-17

skinBundle

activating, 3-10, 4-3

creating, 2-13 to 2-22, 3-18, 4-6

deactivating, 3-42, 4-13

deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20

directory structure, 2-16

exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30

object description, 2-66

returning current state, 3-66, 3-79, 3-90, 4-37, 4-40, 4-43

returning object keys, 3-57, 4-33

updating, 3-109, 4-53

source

creating, 3-19, 3-31, 4-6, 4-9

deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20

exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30

federated, 2-69 to 2-71

file, 2-72

object description, 2-68

returning object keys, 3-57, 4-33

updating, 3-110, 3-123, 4-53, 4-56

Web, 2-82

source code, Java, A-1

source groups, *See* sourceGroup

source types

list of user-defined, 2-78

See also sourceType

sourceGroup

creating, 3-20, 3-32, 4-6, 4-9

deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20

exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30

- object description, 2-93
- returning object keys, 3-57, 4-33
- updating, 3-111, 3-124, 4-53, 4-56
- sources
 - search groups, 2-93
 - See also* source, 2-82
 - user-defined, 2-78
- sourceType
 - creating, 3-21, 3-33, 4-6, 4-9
 - deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 - exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 - object description, 2-95
 - returning object keys, 3-57, 4-33
 - updating, 3-112, 3-125, 4-53, 4-56
- space calculator, *See* spaceCalculator
- space management
 - alert, 2-23
 - procedures, 2-12
 - storageArea, 2-102
 - task object, 2-107
- spaceCalculator
 - activating, 3-11, 4-3
 - deactivating, 3-43, 4-13
 - exporting, 3-51, 4-24
 - object description, 2-99
 - returning current state, 3-81, 4-40
 - starting, 2-12, 3-93, 4-48
 - updating, 3-113, 4-53
- spaces, using in commands, 3-3
- special characters, using in commands, 3-3
- spelling correction, 2-53
- start command, 3-2
- start service request, 4-48
- state errors, B-11
- state properties, 1-10, 2-8
- state, getting current, 3-60 to 3-90, 4-37, 4-40, 4-43
- stateful client (Java), 1-6
- stateful mode, 1-5
- stateful objects, 2-9
- stateful sessions, opening, 4-46
- stateless client (Java), 1-6
- stateless mode, 1-5
- State.xsd, 1-10
- statistics collection, 2-53
- statistics log, for searches, 2-53
- status codes, 1-13
- stem operator, 2-106
- stop command, 3-2
- stop service request, 4-50
- storage areas
 - quota enforcement, 2-99
 - size restrictions, 2-23
- storage areas, *See* storageArea
- storageArea
 - creating, 3-22, 3-34, 4-6, 4-9
 - deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 - exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 - object description, 2-102
 - quotas, 2-12
 - returning object keys, 3-57, 4-33

- updating, 3-114, 3-126, 4-53
- style sheets, 2-14, 2-66
- suggested links, *See* suggLink
- suggLink
 - creating, 3-23, 3-35, 4-6, 4-9
 - deleting, 3-44, 3-45, 3-47, 4-15, 4-17, 4-20
 - exporting, 3-49, 3-53, 3-55, 4-24, 4-27, 4-30
 - object description, 2-105
 - returning object keys, 3-57, 4-33
 - updating, 3-115, 3-127, 4-53, 4-56
- support bundles, 2-14
- SYN (synonym), 2-109
- synonyms, 2-108
- system files, 2-103

T

- table display URL, 2-52
- tablespaces, 2-102
- task
 - exporting, 3-49, 4-24
 - object description, 2-107
 - returning current state, 3-82, 4-40
 - starting, 3-94, 4-48
- template files, 2-66
- template library, 2-15
- templates, 2-14
- thesaurus
 - creating, 3-24, 4-6
 - deleting, 3-44, 4-15
 - exporting, 3-49, 4-24
 - object description, 2-108
 - returning object keys, 3-57, 4-33
 - updating, 3-116, 4-53
- thesaurus operator, 2-106
- threads
 - for searching, 2-54
- threads, for crawling, 2-34
- time-out intervals, 2-54
- top *n* results, 2-30
- topic cluster tree, 2-31
- topic clustering, *See* clustering
- translations, 2-6
- trusted entities, *See* proxyLogin

U

- underscores, using in commands, 3-3
- universal types
 - listed, 1-7
- update command, 3-2
- update service request, 4-52
- updateAll command, 3-2
- updateAll service request, 4-56
- URL recommendations, 2-105
- URL rewriter, 2-87
- URLs for display, 2-52
- URLs, nested link limit, 2-35
- user interface customizations, 2-13 to 2-22, 2-66
- user names, 4-46

user-defined directives, 2-14
user-defined sources, 2-78

V

validation, 2-38
view, *See* export

W

Web sites, searching, 2-68
Web sources, 2-82
wildcard characters, 3-3
within operator, 2-106
WSDL address, 1-5

X

XSLT style sheets, 2-58

Y

YUI library, 2-15