Oracle® Secure Enterprise Search

Administration API Guide

11*g* Release 2 (11.2.2.2)

E35071-08

December 2015

Oracle Secure Enterprise Search Administration API Guide, 11*g* Release 2 (11.2.2.2)

E35071-08

Primary Author: Rajesh Bhatiya

Contributor: Arun Kalyaanasundaram, Aswini Krishnan, Ching-tai Chiu, Cindy Hsin, Diego Iglesias, Greg Brunet, Hiroshi Koide, Luke Wang, Marvin Huang, Michael Kuo, Mohammad Faisal, Mohammed Qureshi, Pranav Agarwal, Pranav Singh, Ravi Palakodety, Rishishankar Rengasamy, Sachin Bhatkar, Sana Karam, Santhosh Thomas, Shashi Anand, Shijun Cheng, Stefan Buchta, Steve Yang, Susanta Mohapatra, Takeshi Yabe, Thomas Chang, Tommy Mo, Valarie Moore, Wei We Chen, Yan Zhao, Yaohua Ma

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 1-6
Using the Web Services Java Client 1-6
Java Libraries 1-6
Creating a Stateful Web Services Client 1-6
Creating a Stateless Web Services Client 1-7
Creating an Administrative Object 1-7
Object Types 1-8
Universal Objects 1-8
Creatable Types 1-9
Object Properties 1-10
XML Documents 1-10
Sample XML Document of an Administration Object 1-10
Editing XML Files 1-11
State Properties 1-12
Operations 1-13
Command Syntax 1-13
Using the Message Logs 1-13
Log Files 1-13
Configuration File 1-13

Managing Universal Objects 1-14
Managing Creatable Objects 1-15
Managing Object State 1-15
Status Codes for Bulk Operations 1-15

2 Administration Object Types

Alphabetic List of Administration Object Types 2-1
Document Formats Supported 2-2
Globalization Support 2-4
Product Languages 2-4
Crawlable Documents 2-5
Providing Translations of Object Names 2-7
Encryption 2-9
XML Description of State Properties 2-9
Search Interface Customization: Skin Bundles 2-11
Support Bundles 2-11
FreeMarker Templates 2-11
Asset Files 2-12
JavaScript Libraries 2-12
Template Library 2-13
Assembling the Skin Bundle Files 2-13
Creating a skinBundle Object 2-13
Using a Skin Bundle to Render the Search Application User Interface 2-14
Skin Bundle Example 2-14
Changes to the Example Results Page 2-14
Changes to the Example Footer 2-15
Creating the Example Directory Structure 2-16
Customizing results.ftl 2-16
Customizing inc_logo_querybox.ftl 2-17
Customizing inc_footer.ftl 2-18
Creating the Example Skin Bundle File List 2-18
Creating an XML Description of the Example Skin Bundle 2-18
Creating the Example skinBundle Object 2-19
Using the Example Skin Bundle to Render the Search Application 2-19
altWord 2-20
authorizedPrincipal 2-22
autoSuggestion 2-24
boostedUrl 2-27
classification 2-29
classificationMappings 2-31
clustering 2-33
clusterTree 2-36
crawlerSettings 2-40
docServiceInstance 2-45
docServiceManager 2-48
docServicePipeline 2-51
facetTree 2-54

globalBoundaryRules 2-60
globalDocumentTypes 2-63
identityPlugin 2-65
index 2-69
indexOptimizer 2-71
indexProfile 2-74
languageBasedTokenization 2-77
lexer 2-79
partitionConfig 2-87
proxy 2-92
proxyLogin 2-94
queryConfig 2-96
queryUIConfig 2-102
queryUIFacets 2-108
queryUISourceGroups 2-110
relevanceRanking 2-115
resultList 2-123
schedule 2-125
searchAttr 2-129
singleSignOnSetting 2-132
skinBundle 2-135
source 2-137
sourceGroup 2-165
sourceType 2-167
storageArea 2-171
suggContent 2-173
suggContentProvider 2-174
suggestion 2-179
suggLink 2-181
tagging 2-183
tag 2-185
thesaurus 2-187

3 searchadmin Commands

Alphabetic List of searchadmin Commands 3-1
activate clustering 3-5
activate clusterTree 3-6
activate identityPlugin 3-7
activate indexOptimizer 3-8
activate partitionConfig 3-9
activate queryUIFacets 3-10
activate resultList 3-11
activate schedule 3-12
activate singleSignOnSetting 3-13
activate skinBundle 3-14
activate suggContentProvider 3-15
activate tagging 3-16

create altWord 3-17
create boostedUrl 3-18
create classification 3-19
create clusterTree 3-20
create docServiceInstance 3-21
create docServiceManager 3-22
create docServicePipeline 3-23
create facetTree 3-24
create identityPlugin 3-25
create indexProfile 3-26
create lexer 3-27
create proxyLogin 3-28
create schedule 3-29
create searchAttr 3-30
create skinBundle 3-31
create source 3-32
create sourceGroup 3-33
create sourceType 3-34
create storageArea 3-35
create suggContentProvider 3-36
create suggestion 3-37
create suggLink 3-38
create tag 3-39
create thesaurus 3-40
createAll altWord 3-41
createAll boostedUrl 3-42
createAll authorizedPrincipal 3-43
createAll classification 3-44
createAll clusterTree 3-45
createAll docServiceInstance 3-46
createAll docServiceManager 3-47
createAll docServicePipeline 3-48
createAll facetTree 3-49
createAll identityPlugin 3-50
createAll indexProfile 3-51
createAll lexer 3-52
createAll proxyLogin 3-53
createAll schedule 3-54
createAll searchAttr 3-55
createAll source 3-56
createAll sourceGroup 3-57
createAll sourceType 3-58
createAll storageArea 3-59
createAll suggContentProvider 3-60
createAll suggestion 3-61
createAll suggLink 3-62
createAll tag 3-63

deactivate clustering 3-64
deactivate clusterTree 3-65
deactivate identityPlugin 3-66
deactivate indexOptimizer 3-67
deactivate queryUIFacets 3-68
deactivate resultList 3-69
deactivate schedule 3-70
deactivate singleSignOnSetting 3-71
deactivate skinBundle 3-72
deactivate suggContentProvider 3-73
deactivate tagging 3-74
delete ***creatable_type*** 3-75
deleteAll ***creatable_type*** 3-76
deleteList ***creatable_type*** 3-78
export ***creatable_type*** 3-80
export ***universal_type*** 3-82
exportAll ***creatable_type*** 3-84
exportList ***creatable_type*** 3-86
getAllObjectKeys 3-88
getAPIVersion 3-90
getAllStates clusterTree 3-91
getAllStates identityPlugin 3-93
getAllStates schedule 3-95
getAllStates singleSignOnSetting 3-97
getAllStates skinBundle 3-99
getAllStates suggContentProvider 3-101
getState autoSuggestion 3-103
getState clustering 3-104
getState clusterTree 3-105
getState identityPlugin 3-107
getState index 3-109
getState indexOptimizer 3-110
getState partitionConfig 3-112
getState queryUIFacets 3-113
getState resultList 3-114
getState schedule 3-115
getState singleSignOnSetting 3-117
getState skinBundle 3-119
getState suggContentProvider 3-121
getState tagging 3-123
getStateList clusterTree 3-124
getStateList identityPlugin 3-126
getStateList schedule 3-128
getStateList singleSignOnSetting 3-130
getStateList skinBundle 3-131
getStateList suggContentProvider 3-132
start autoSuggestion 3-134

start indexOptimizer 3-135
start schedule 3-136
stop indexOptimizer 3-137
stop schedule 3-138
update altWord 3-139
update autoSuggestion 3-140
update boostedUrl 3-141
update classification 3-142
update classificationMappings 3-143
update clustering 3-144
update clusterTree 3-145
update crawlerSettings 3-146
update docServiceInstance 3-147
update docServicePipeline 3-148
update facetTree 3-149
update globalBoundaryRules 3-150
update globalDocumentTypes 3-151
update index 3-152
update indexOptimizer 3-153
update indexProfile 3-154
update languageBasedTokenization 3-155
update lexer 3-156
update partitionConfig 3-157
update proxy 3-158
update proxyLogin 3-159
update queryConfig 3-160
update queryUIConfig 3-161
update queryUIFacets 3-162
update queryUISourceGroups 3-163
update relevanceRanking 3-164
update resultList 3-165
update schedule 3-166
update searchAttr 3-167
update singleSignOnSetting 3-168
update skinBundle 3-169
update source 3-170
update sourceGroup 3-171
update sourceType 3-172
update storageArea 3-173
update suggContent 3-174
update suggContentProvider 3-175
update suggestion 3-176
update suggLink 3-177
update tag 3-178
update tagging 3-179
update thesaurus 3-180
updateAll altWord 3-181

updateAll boostedUrl 3-182
updateAll classification 3-183
updateAll clusterTree 3-184
updateAll docServiceInstance 3-185
updateAll docServicePipeline 3-186
updateAll facetTree 3-187
updateAll indexProfile 3-188
updateAll lexer 3-189
updateAll proxyLogin 3-190
updateAll schedule 3-191
updateAll searchAttr 3-192
updateAll schedule 3-194
updateAll singleSignOnSetting 3-195
updateAll sourceGroup 3-196
updateAll sourceType 3-197
updateAll storageArea 3-198
updateAll suggContentProvider 3-199
updateAll suggestion 3-200
updateAll suggLink 3-201
updateAll tag 3-202

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 4-18
deleteList 4-21
export 4-25
exportAll 4-28
exportList 4-31
getAllObjectKeys 4-35
getAPIVersion 4-37
getAllStates 4-39
getState 4-42
getStateList 4-45
login 4-48
logout 4-49
start 4-50
stop 4-52
update 4-54
updateAll 4-58

A Java Example

Java Source Code Example A-1
Shell Script Example A-5

B Error Messages

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

Index

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/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` 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/technetwork/community/join/overview/index.htm`

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/technetwork/indexes/documentation/index.html`

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.
italic	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
italics	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.
oracle_base	*oracle_base* is the root of the directory structure where all the Oracle Database products are installed.
oracle_home	*oracle_home* refers to the directory where the database component specific to Oracle SES software is installed. The default *oracle_home* directory is *oracle_base*/product/11.2.0/sesdb_1.
mw_home	*mw_home* refers to the middleware home directory, where the middle tier component of Oracle SES software (that is, Oracle WebLogic Server, Oracle SES server, Oracle Fusion Middleware common files) is installed.
ses_home	*ses_home* refers to the directory where the Oracle SES server is installed. The *ses_home* directory is created under the *mw_home* directory. When Oracle SES 11.2.2.2 software is installed along with the WebLogic Server middle tier, the *ses_home* directory is *mw_home*/Oracle_SES1.
wls_domain_home	*wls_domain_home* refers to the directory where multiple instances of a WebLogic Server middle tier are created. The default *wls_domain_home* directory is *mw_home*/user_projects/domains.
ses_domain_name	*ses_domain_name* refers to the directory where the WebLogic Server instance specific to Oracle SES is created under *wls_domain_home* directory. The default *ses_domain_name* directory is search_domain.
java_home	*java_home* refers to the directory where JDK is installed in the Oracle SES middle tier. When Oracle SES 11.2.2.2 software is installed along with the WebLogic Server middle tier, the *java_home* directory is *ses_home*/jdk.

1 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 *`ses_home`*`/bin/searchadmin` command at the operating system prompt. You can connect to a single instance or multiple instances.

To connect to a single instance, use the `--CONNECTION` option with an HTTP connection string, like the one shown here.

```
$ searchadmin --CONNECTION=http://ses_host:ses_port/search/api/admin/AdminService
```

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

Search Admin Command Line - Release 11.2.2.2.0

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

Password:

SES>getAPIVersion

11.2.2.2.0

SES>export index

<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
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 --CONNECTION=http://myhost:7777/search/api/admin/AdminService
getAPIVersion

Search Admin Command Line - Release 11.2.2.2.0
```

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

Password:

11.2.2.2.0

$ searchadmin --CONNECTION=http://myhost:7777/search/api/admin/AdminService
--PASSWORD=password export index

Search Admin Command Line - Release 11.2.2.2.0

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


<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
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 search instance defined by the CONNECTION parameter. However, you can issue commands to multiple Oracle SES instances by supplying the connection information in a text file. Then reference the file using the `--CONNECTION_LIST` option. Each command is executed across all instances.

For example, create a file named `remotehosts.lst` and enter the information related to multiple instances in it as shown below:

```
--USERNAME=searchsys --CONNECTION=http://ses_host_1:ses_port_
1/search/api/admin/AdminService

--USERNAME=searchsys --CONNECTION=http://ses_host_2:ses_port_
2/search/api/admin/AdminService
```

This command opens connections using the `remotehosts.lst` file:

```
$ 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 11.2.2.2.0

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 searchadmin command syntax, and lists the administrative object types and operations.
help *object_type*	Describes *object_type*.
help *operation*	Provides the command syntax for *operation*.
help statusCodes	Describes the status codes that are displayed after an operation is performed on multiple objects by createAll, deleteAll, deleteList, or updateAll.
help duplicateMethod	Describes the settings of the --DUPE_METHOD parameter in the createAll command.
help updateMethod	Describes the settings of the --UPDATE_METHOD parameter in the update and updateAll commands.
help notFoundMethod	Describes the settings of the --NOT_FOUND_METHOD parameter in the updateAll 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 searchsys 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 an Oracle SES instance using HTTP:

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

The port number is the same as connecting to 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.

Note: To enter special characters as part of a `searchadmin` command, you may have to enclose the value in quotes, or prefix the character with backslashes as escape characters, or use both.

- If the command argument value contains spaces, then enclose it in double quotes. For example: `--NAME="Special Collection"`.
- To use the percent (`%`) and underscore (_) wildcard characters as literals for key patterns in operations such as `createAll` and `deleteAll`, prefix the wildcard character with two backslashes (\\) on a Linux or a UNIX system and prefix it with a single backslash (\) on a Windows system.

  For example, on a Linux or a UNIX system, escape the wildcard character _ in the object named `web_source` using the syntax `--NAME=web\\_source`. The same functionality can be achieved on a Windows system using the syntax `--NAME=web\_source`.

  In single job mode, also enclose the values containing these characters in double quotes. For example, on a Linux or a UNIX system, use the syntax `--NAME="web\\_source"` and on a Windows system, use the syntax `--NAME="web\_source"`.
- 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`. On Windows system, always use the long form for arguments in single job mode as well as in session mode.

See "Command Interface" on page 1-1 for a discussion of session mode and single job mode.

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

You need to include the following JAR files in the CLASSPATH for using the Oracle SES Web services Java client:

- *`mw_home`*`/oracle_common/modules/oracle.webservices_11.1.1/oracle.webservices.standalone.client.jar`
- *`mw_home`*`/search/lib/search_adminapi_wsclient.jar`
- For the IBM-AIX platform, you also require the JAR file:

  `mw_home``/oracle_common/modules/glassfish.jaxb.xjc.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=\"11.2.2.2.0\"
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
autoSuggestion	Auto suggestions configuration
classificationMappings	Group specific classification mappings for suggestion keywords
clustering	Clustering configuration
crawlerSettings	Crawler configuration
globalBoundaryRules	Global crawler boundary rules
globalDocumentTypes	Global crawler document types
index	Indexing parameters
indexOptimizer	Index optimization
languageBasedTokenization	Language specific lexers
partitionConfig	Partition configuration
proxy	HTTP proxy server configuration
queryConfig	Query configuration
queryUIConfig	Query UI configuration
queryUIFacets	Facets configuration
queryUISourceGroups	Source group configuration
relevanceRanking	Attribute relevance ranking

Table 1–2 (Cont.) Universal Objects

Object Type	Description
resultList	Search result list configuration
suggContent	Suggested content configuration
tagging	Tagging configuration

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
altWord	Alternate words
authorizedPrincipal	Authorizations for functinality, such as, tagging, to various Oracle SES users
boostedUrl	Relevancy boosting URLs
classification	Classification configuration for categorizing suggestion keywords
clusterTree	Cluster trees
docServiceInstance	Document service instance
docServiceManager	Document service manager
docServicePipeline	Document service pipeline
facetTree	Facet trees
identityPlugin	Identity plug-ins
indexProfile	Index profiles
lexer	Lexers
proxyLogin	Proxy log-ins
schedule	Schedules
searchAttr	Search attributes
singleSignOnSetting	Single sign-on configuration
skinBundle	Skin bundle
source	Sources
sourceGroup	Source groups

Table 1–3 (Cont.) Creatable Object Types

Object Type	Description
`sourceType`	Source types
`storageArea`	Storage areas
`suggContentProvider`	Suggested content provider
`suggestion`	Suggestions
`suggLink`	Suggested links
`tag`	Tags for bulk upload
`thesaurus`	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="11.2.2.2.0"
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. It contains the `productVersion` attribute that denotes the Oracle SES product version.

Note: The `productVersion` attribute is optional, but it is strongly recommended to specify it, because different versions of Oracle SES may have behavioral differences in their Admin API. When the `productVersion` is not specified, Oracle SES assumes it to be the latest available version, that is, 11.2.2.2.

In case the `productVersion` is specified as 11.1.2.2 while importing an XML into an Oracle SES 11.2.2.2 instance, then Oracle SES will use the **11.1.2.2** version behavior.

For example, the earlier Oracle SES release 11.1.2.2 did not support master encryption key, but the current Oracle SES release 11.2.2.2 supports it. Thus, to provide backward compatibility, Oracle SES 11.2.2.2 handles the 11.1.2.2 encryption format correctly whenever the `productVersion` attribute value is specified as 11.1.2.2.0 in the input XML.

<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 the *`ses_home`*`/search/xsd/admin/api` directory.

Note:

- Oracle SES automatically encrypts any secure information stored in it, such as password, when it is exported to a text file. For example:

```
<search:parameter name="Password">
   <search:value
encrypted="true">dc33fb22cfaf96fd592b14754f58818911acaa08e98296
6f</search:value>
   <search:description>Service Operation User
password</search:description>
</search:parameter>
```

- When using a configuration file stored on a disk, any security information in it, such as password, should be typed in plain text with the `encrypted` property value set to `false`. For example:

```
<search:value encrypted="false">not4U2Know</search:value>
```

- A configuration file containing unencrypted security information should be protected, or else it should be removed immediately after use.

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 2-9.

Table 1–4 Administrative Objects With State Properties

Object Type	State Properties
`autoSuggestion`	`filterStatus, filterError`
`clustering`	`status`
`clusterTree`	`status`
`identityPlugin`	`status`
`index`	`estimatedFragmentation`
`indexOptimizer`	`endTime`, `startTime`, `status`
`partitionConfig`	`status`
`queryUIFacets`	`status`
`resultList`	`status`
`schedule`	`lastCrawled`, `logFilePath`, `nextCrawl`, `scheduleError`, `status`
`skinBundle`	`status`
`suggContentProvider`	`status`

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 stores log messages in the directory:

wls_domain_home/*ses_domain_name*/servers/search_server1/logs

The WebLogic server for Web services stores log messages in the directory:

wls_domain_home/*ses_domain_name*/servers/AdminServer/logs

Configuration File

Oracle SES 11.2.2.2 uses ODL as the default logger. The earlier Oracle SES release (11.1.2.2) used log4j as the default logger. Oracle SES 11.2.2.2 stores logger configurations for ODL logs in the script file *ses_home*/bin/clexecutor.sh. You can update this file to configure the following parameters for the logger:

- LOG_FORMAT: The log format to use. The available values are: ODL-XML, ODL10-XML, and ODL-Text. Default is ODL-Text.
- LOG_MAX_FILE_SIZE: The maximum size in bytes for each log file. When a log file reaches this size, it is archived, and a new log file is created. Default is 104857600 bytes.
- LOG_ROTATION_FREQUENCY: The frequency, in minutes, for archiving the old log file and creating a new one. This value must be either a number (that is, minutes), or one of the following values (case-insensitive): hourly, daily, or weekly. Default is daily.

> **Note:** If a value is specified for the LOG_MAX_FILE_SIZE parameter, then Oracle SES ignores the value specified for the LOG_ROTATION_FREQUENCY parameter.

- LOG_LEVEL: Sets the log level in the form of a number. ODL logs only those records with a level equal to or higher than the specified log level value. The meaningful name for each log level is as follows:

Log Level	Name
0	TRACE
2	DEBUG
4	INFO (NOTIFICATION)
6	WARN
8	ERROR (SEVERE)
10	FATAL (INCIDENT_ERROR)

- LOG_AUTOFLUSH_LEVEL: Sets the log level for auto-flushing. The ODL allows log records to be buffered, till it encounters a log record with a level equal to or higher than the specified auto-flush level, and at that time ODL automatically flushes the buffer. The available values are: TRACE, DEBUG, NOTIFICATION, WARN, ERROR, and FATAL. Default is NOTIFICATION.

> **Note:** The log level can also be changed using the Administration API (<logLevel> child element of the <crawlerSettings> element of the sources object). Oracle SES uses the log level that is more granular of the two (configured in clexecutor.sh file and configured using Administration API).

Managing Universal Objects

Use these operations to manage universal administration objects:

Table 1–5 Operations on Universal Objects

Operation	Description
export	Returns the XML description of an object.

Table 1–5 (Cont.) Operations on Universal Objects

Operation	Description
`update`	Sets the parameters of an object from an XML file.

Managing Creatable Objects

Use these operations to manage creatable administration objects:

Table 1–6 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 1–7 Operations on Objects With State Properties

Operation	Description
`activate`	Enables an object. Objects that can be enabled have a `status` state property.
`deactivate`	Disables an object. Objects that can be disabled have a `status` 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 1–8.

Table 1–8 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.

2

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 Formats Supported
- Globalization Support
- Encryption
- XML Description of State Properties
- Search Interface Customization: Skin Bundles

Alphabetic List of Administration Object Types

A B C D F G I L P Q R S T

A

`altWord`
`authorizedPrincipal`
`autoSuggestion`

B

`boostedUrl`

C

`classification`
`classificationMappings`
`clustering`
`clusterTree`
`crawlerSettings`

D

`docServiceInstance`
`docServiceManager`
`docServicePipeline`

F

`facetTree`

G

globalBoundaryRules
globalDocumentTypes

I

identityPlugin
index
indexOptimizer
indexProfile

L

languageBasedTokenization
lexer

P

partitionConfig
proxy
proxyLogin

Q

queryConfig
queryUIConfig
queryUIFacets
queryUISourceGroups

R

relevanceRanking
resultList

S

schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContent
suggContentProvider
suggestion
suggLink

T

tagging
tag
thesaurus

Document Formats Supported

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

Table 2–1 Document Formats Supported by Oracle SES

Document Format	MIME Type
Adobe Framemaker Interchange Format (MIF) Document	application/vnd.mif
Corel Presentations Document	application/vnd.corel-presentations
DICOM Image	application/dicom
GIF Image	image/gif
GNU ZIP Archive	application/x-gzip
Haansoft Hangul Document	application/x-hwp
HTML	text/html
JPEG 2000 Image	image/jp2
JPEG Image	image/jpeg
JustSystems Ichitaro	application/x-js-taro
Lotus 1-2-3 Document	application/x-lotus123 application/vnd.lotus-1-2-3
Lotus Freelance Document	application/x-freelance application/vnd.lotus-freelance
Lotus Word Pro Document	application/vnd.lotus-wordpro
LHA Archive	application/x-lzh-compressed
Microsoft Excel Document	application/x-msexcel application/vnd.ms-excel, application/ms-excel application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
Microsoft Project Document	application/vnd.ms-project
Microsoft PowerPoint Document	application/x-mspowerpoint application/vnd.ms-powerpoint application/vnd.openxmlformats-officedocument.presentationml.presentation
Microsoft Visio Document	application/vnd.visio
Microsoft Word Document	application/msword application/vnd.openxmlformats-officedocument.wordprocessingml.document
Microsoft Works Word Processor Document	application/x-msworks-wp
MS Write	application/x-mswrite
PDF Document	application/pdf
Plain Text	text/plain
PostScript Document	application/postscript, application/ps, application/x-postscript, application/x-ps
Quattro Pro for Windows Document	application/x-quattro-win
Rich Text Format (RTF) Document	application/rtf

Table 2–1 (Cont.) Document Formats Supported by Oracle SES

Document Format	MIME Type
StarOffice/OpenOffice Calc Document	application/vnd.stardivision.calc
StarOffice/OpenOffice Impress Document	application/vnd.stardivision.impress
StarOffice/OpenOffice Draw Document	application/vnd.stardivision.draw
StarOffice/OpenOffice Writer Document	application/vnd.stardivision.writer
TIF Image	image/tiff
WordPerfect 5.1 Document	application/wordperfect5.1
WordPerfect 6 Document	application/x-wordperfect6
X-Ami Document	application/amipro, application/x-amipro, application/sam, application/x-sam application/x-ami
XML	text/xml
XyWrite Document	application/x-xywrite
ZIP Archive	application/zip

Note: Oracle SES uses Oracle Text to convert binary documents to HTML. See "Appendix B" of *Oracle Text Reference* for more information about the document formats supported by Oracle SES.

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 Languages Supported by the Crawler

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
suggContentProvider
```

<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.

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:

```
wls_domain_home/ses_domain_name/servers/search_server1/tmp/_WL_user/search_
query/curkae/war/WEB-INF/templates
```

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:

```
<img src="<@ses.skin_asset filename='images/mylogo.gif'/>" ALT="Example, Inc."/>
```

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 11.2.2.2.0 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://sourceforge.net/projects/bubbling/`

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.

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 2-11.
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 2-135.

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 3-31.
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

`wls_domain_home``/`*`ses_domain_name`*`/servers/AdminServer/logs/AdminServer.out`

After updating the skin bundle, restart the middle tier.

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.

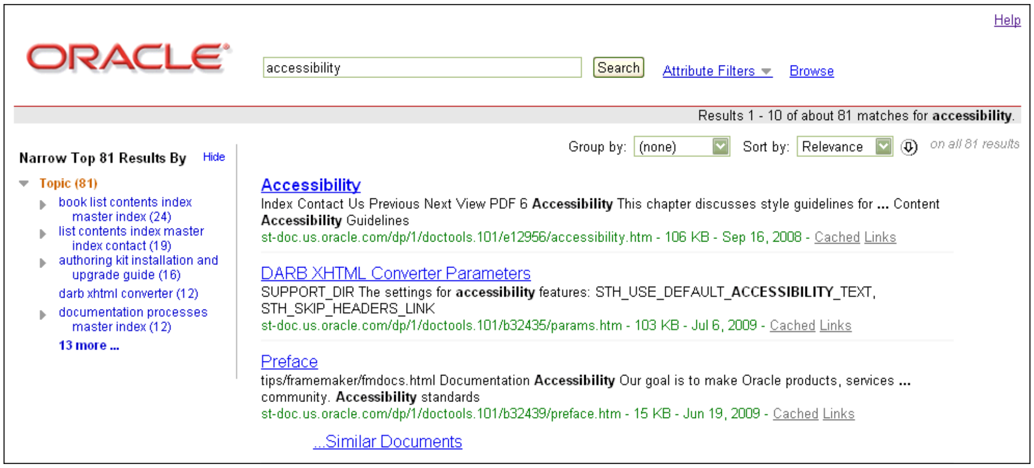
Changes to results.ftl do not affect any other pages of the Search application, which continue to use the default skin. However, the example makes changes to inc_logo_querybox.ftl and inc_footer.ftl, which affect all of the pages that include those templates.

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	inc_logo_querybox.ftl
Search button	Search icon	inc_logo_querybox.ftl
Sidebar on left	Sidebar on right	results.ftl

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



Help

ORACLE

accessibility Search Attribute Filters Browse

Results 1 - 10 of about 81 matches for **accessibility**.

Narrow Top 81 Results By Hide

Topic (81)
- book list contents index master index (24)
- list contents index master index contact (19)
- authoring kit installation and upgrade guide (16)
- darb xhtml converter (12)
- documentation processes master index (12)
- 13 more ...

Group by: (none) Sort by: Relevance *on all 81 results*

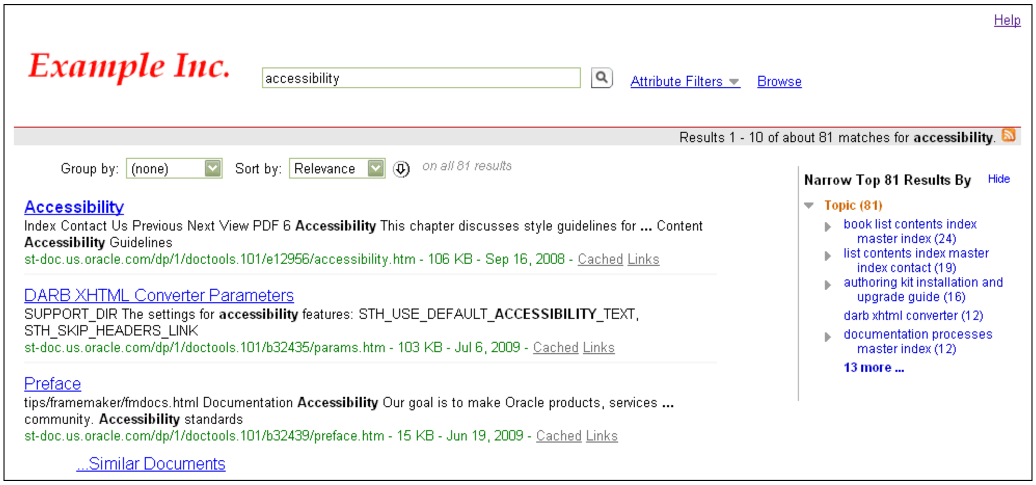
Accessibility
Index Contact Us Previous Next View PDF 6 **Accessibility** This chapter discusses style guidelines for **...** Content **Accessibility** Guidelines
st-doc.us.oracle.com/dp/1/doctools.101/e12956/accessibility.htm - 106 KB - Sep 16, 2008 - Cached Links

DARB XHTML Converter Parameters
SUPPORT_DIR The settings for **accessibility** features: STH_USE_DEFAULT_**ACCESSIBILITY**_TEXT, STH_SKIP_HEADERS_LINK
st-doc.us.oracle.com/dp/1/doctools.101/b32435/params.htm - 103 KB - Jul 6, 2009 - Cached Links

Preface
tips/framemaker/fmdocs.html Documentation **Accessibility** Our goal is to make Oracle products, services **...** community. **Accessibility** standards
st-doc.us.oracle.com/dp/1/doctools.101/b32439/preface.htm - 15 KB - Jun 19, 2009 - Cached Links

...Similar Documents

Figure 2–1 Default Results Page



Help

Example Inc.

accessibility Attribute Filters Browse

Results 1 - 10 of about 81 matches for **accessibility**.

Group by: (none) Sort by: Relevance *on all 81 results*

Accessibility
Index Contact Us Previous Next View PDF 6 **Accessibility** This chapter discusses style guidelines for **...** Content **Accessibility** Guidelines
st-doc.us.oracle.com/dp/1/doctools.101/e12956/accessibility.htm - 106 KB - Sep 16, 2008 - Cached Links

DARB XHTML Converter Parameters
SUPPORT_DIR The settings for **accessibility** features: STH_USE_DEFAULT_**ACCESSIBILITY**_TEXT, STH_SKIP_HEADERS_LINK
st-doc.us.oracle.com/dp/1/doctools.101/b32435/params.htm - 103 KB - Jul 6, 2009 - Cached Links

Preface
tips/framemaker/fmdocs.html Documentation **Accessibility** Our goal is to make Oracle products, services **...** community. **Accessibility** standards
st-doc.us.oracle.com/dp/1/doctools.101/b32439/preface.htm - 15 KB - Jun 19, 2009 - Cached Links

...Similar Documents

Narrow Top 81 Results By Hide

Topic (81)
- book list contents index master index (24)
- list contents index master index contact (19)
- authoring kit installation and upgrade guide (16)
- darb xhtml converter (12)
- documentation processes master index (12)
- 13 more ...

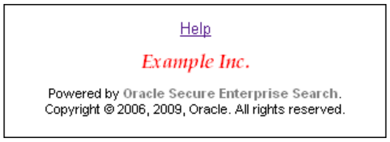
Figure 2–2 Example Results Page

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



Help

Example Inc.

Powered by Oracle Secure Enterprise Search.
Copyright © 2006, 2009, Oracle. All rights reserved.

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:

   ```
   wls_domain_home/ses_domain_name/servers/search_server1/tmp/_WL_user/search_
   query/curkae/war/WEB-INF/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

```
<img src="<@ses.skin_asset filename='images/example.gif'/>">
```

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="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:skinBundles>
      <search:skinBundle>
         <search:name>example</search:name>
         <search:isDefault>false</search:isDefault>
         <search:linkedVersion>11.2.2.2.0</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".

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="11.2.2.2.0"
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>
```

authorizedPrincipal

The `authorizedPrincipal` object is used to provide various privileges, such as tagging, to the required Oracle SES users.

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

The `<search:authorizedPrincipals>` element describes privileges for Oracle SES users:

```
<search:authorizedPrincipals>
  <search:authorizedPrincipal>
    <search:name>
    <search:privileges>
      <search:privilege>
```

Element Descriptions

<search:authorizedPrincipals>
Contains one or more <search:authorizedPrincipal> elements.

<search:authorizedPrincipal>

Describes the previleges for a user. It contains these elements:

```
<search:name>
<search:privileges>
```

<search:name>

Name of the user to whom the required privileges are assigned using the `<search:privileges>` element.

<search:privileges>

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

<search:privilege>

Describes a privilege for a user.

Attribute	Value
`type`	Type of privilege, such as `TAGGING`. Required.

Example

This example assigns `tagging` privilege to users - `user1` and `user2`:

```
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:authorizedPrincipals>
      <search:authorizedPrincipal>
         <search:name>user1</search:name>
         <search:privileges>
            <search:privilege type="TAGGING"/>
         </search:privileges>
      </search:authorizedPrincipal>
      <search:authorizedPrincipal>
         <search:name>user2</search:name>
         <search:privileges>
            <search:privilege type="TAGGING"/>
         </search:privileges>
      </search:authorizedPrincipal>
   </search:authorizedPrincipals>
</search:config>
```

autoSuggestion

The autoSuggestion object is used to configure general settings for auto suggestions.

Object Type

Universal

State Properties

Property	Value
filterStatus	ACTIVE INACTIVE
filterError	An error value is assigned to this property by Oracle SES in case of any error while processing auto suggestions.

Supported Operations

```
export
getState
update
start
```

Administration GUI Page

None

XML Description

The `<search:autoSuggestions>` element describes auto suggestion configurations:

```
<search:autoSuggestion>
   <search:maxSuggestions>
   <search:maxTotalSuggestions>
   <search:maxSuggestionLength>
   <search:populateFromQueries>
      <search:minOccurrenceOfPhrase>
      <search:filterExpression>
      <search:populateFromSecureQueries>
         <search:minDistinctUsers>
```

Element Descriptions

<search:autoSuggestion>
Contains these elements:

```
<search:maxSuggestions>
<search:maxTotalSuggestions>
<search:maxSuggestionLength>
<search:populateFromQueries>
```

<search:maxSuggestions>
The maximum number of suggestion keywords to display in the search box of the query application. It must be a numeric value greater than 0 and less than 16.

<search:maxTotalSuggestions>

The maximum number of suggestion keywords to store in Oracle SES.

<search:maxSuggestionLength>

The maximum length of a suggestion keyword.

<search:populateFromQueries>

Contains these elements:

```
<search:minOccurenceOfPhrase>
<search:filterExpression>
<search:populateFromSecureQueries>
```

Controls whether the auto suggestion keywords are populated from queries.

Attribute	Value
`enabled`	Set to `true` to populate auto suggestion keywords from queries, or set to `false` otherwise. Required.

<search:minOccurrenceOfPhrase>

The minimum number of times a phrase must be searched so as to add it to the suggestion keywords list while populating the list from queries.

<search:filterExpression>

A regular expression for filtering out undesired keywords from the suggestion keywords list while populating the list from queries.

<search:populateFromSecureQueries>

Contains the `<search:minDistinctUsers>` element.

Controls whether the keywords are populated from secure queries.

Attribute	Value
`enabled`	Set to `true` to populate auto suggestion keywords from query log for secure queries, or set to `false` otherwise. Required.

<search:minDistinctUsers>

The minimum number of distinct users that must search for a phrase in order for that phrase to be added to the suggestion keywords list while populating the list from secure queries.

Example

This XML document configures auto suggestions in Oracle SES:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:autoSuggestion>
      <search:maxSuggestions>15</search:maxSuggestions>
      <search:maxTotalSuggestions>1000000</search:maxTotalSuggestions>
      <search:maxSuggestionLength>60</search:maxSuggestionLength>
      <search:populateFromQueries enabled="true">
         <search:minOccurrenceOfPhrase>100</search:minOccurrenceOfPhrase>
         <search:populateFromSecureQueries enabled="true">
            <search:minDistinctUsers>25</search:minDistinctUsers>
         </search:populateFromSecureQueries>
```

```
        </search:populateFromQueries>
    </search:autoSuggestion>
</search:config>
```

boostedUrl

The boostedUrl object is used to increase the relevancy of specific URLs so that those URLs are displayed on the top of search results.

Object Type

Creatable

Object Key

```
docUrl query
```

Object Key Command Syntax

```
--DOC_URL=url --QUERY=query_term
```

State Properties

None

Supported Operations

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

Administration GUI Page

Search - Relevancy

XML Description

The `<search:boostedUrl>` element describes configurations related to relevancy boosting for specific URLs:

```
<search:boostedUrls>
  <search:boostedUrl>
    <search:url>
    <search:query>
    <search:score>
```

Element Descriptions

<search:boostedUrls>
Contains one or more <search:boostedUrl> elements.

<search:boostedUrl>
Describes the boosted score for a URL. It contains these elements:

```
<search:url>
<search:query>
<search:score>
```

<search:url>

Contains the valid URL of a document whose relevancy requires boosting.

<search:query>

Contains the query term for which `<search:url>` is boosted. Oracle SES requires an exact match for boosting.

<search:score>

Contains an integer from 0 to 100 for the score. Boosted documents are listed in descending order in the search results, before the unboosted documents.

Example

This example boosts two URLs for the search term "indexing":

```
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
<search:boostedUrls>
  <search:boostedUrl>
    <search:url>
       http://example.com/doctools/b32440/xref_foot_in.htm
    </search:url>
    <search:query>indexing</search:query>
    <search:score>90</search:score>
  </search:boostedUrl>

  <search:boostedUrl>
    <search:url>
       http://example.com/doctools/b32439/markers.htm
    </search:url>
    <search:query>indexing</search:query>
    <search:score>80</search:score>
  </search:boostedUrl>
</search:boostedUrls>
</search:config>
```

classification

The classification object is used to specify classifications for categorizing suggestion keywords.

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

The `<search:classifications>` element contains classification details:

```
<search:classifications>
   <search:classification>
      <search:name>
      <search:description>
```

Element Descriptions

<search:classifications>
Contains one or more `<search:classification>` elements.

<search:classification>
Describes a classification. Contains the following elements:

```
<search:name>
<search:description>
```

<search:name>
Name of the classification.

<search:description>
Description of the classification.

Example

This XML document configures classification:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:classifications>
      <search:classification>
         <search:name>Reports</search:name>
         <search:description>Generic Reports</search:description>
      </search:classification>
   </search:classifications>
</search:config>
```

classificationMappings

The classificationMappings object is used to specify source group specific classifications, thus determining the categories of suggestion keywords that are available for each source group.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

None

XML Description

The `<search:classificationMappings>` element describes details related to the mappings between source groups and classifications:

```
<search:classificationMappings>
   <search:classificationMapping>
      <search:sourceGroup>
      <search:classifications>
      <search:classification>
```

Element Descriptions

<search:classificationMappings>
Contains one or more `<search:classificationMapping>` elements.

<search:classificationMapping>
Describes a classification mapping. Contains the following elements:

```
<search:source>
<search:classifications>
```

<search:sourceGroup>
Describes the source group.

Attribute	Value
`name`	Name of the source group.

<search:classifications>
Contains one or more `<search:classification>` elements mapped to the source group.

<search:classification>
Name of the classification.

Attribute	Value
priority	Specifies the priority of the classification. A classification with priority of 1 has higher precedence over a classification with priority of 2.

Example

This XML document configures a source group specific classification:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:classificationMappings>
      <search:classificationMapping>
         <search:sourceGroup name="people"/>
         <search:classifications>
            <search:classification priority="1">Names</search:classification>
            <search:classification priority="2">Addresses</search:classification>
         </search:classifications>
      </search:classificationMapping>
   </search:classificationMappings>
</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 configurations related to clustering:

```
<search:clustering>
   <search:maxTreeDepth>
   <search:maxChildrenPerNode>
   <search:minDocsPerNode>
   <search:minOccurrenceWords>
   <search:maxExtractWords>
   <search:minOccurrencePhrases>
   <search:maxExtractPhrases>
   <search:maxPhraseLength>
   <search:numFirstLevelNode>
   <search:showEmptyCluster>
   <search:topic>
   <search:metaData>
```

Element Descriptions

<search:clustering>
Contains the elements for clustering parameters that 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.

<search:numFirstLevelNode>
Number of cluster nodes to display in the first level of a cluster tree on the search results page.

<search:showEmptyCluster>
Controls whether to show empty clusters on the search results page.

Attribute	Value
`enabled`	Set to `true` to show empty clusters on the search results page, or set to `false` otherwise. Required.

<search:topic>
Settings related to `topic` cluster trees only. It contains the elements `<maxTreeDepth>`, `<maxChildrenPerNode>`, and `<minDocsPerNode>`, which when specified, override the corresponding global settings.

<search:metaData>
Settings related to `metaData` cluster trees only. It contains the elements `<maxTreeDepth>`, `<maxChildrenPerNode>`, and `<minDocsPerNode>`, which when specified, override the corresponding global settings.

Example

This XML document configures clustering:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
```

```
   <search:clustering>
      <search:maxTreeDepth>4</search:maxTreeDepth>
      <search:maxChildrenPerNode>50</search:maxChildrenPerNode>
      <search:minDocsPerNode>3</search:minDocsPerNode>
      <search:minOccurrenceWords>3</search:minOccurrenceWords>
      <search:maxExtractWords>20</search:maxExtractWords>
      <search:minOccurrencePhrases>2</search:minOccurrencePhrases>
      <search:maxExtractPhrases>10</search:maxExtractPhrases>
      <search:maxPhraseLength>6</search:maxPhraseLength>
      <search:numFirstLevelNode>5</search:clusterMoreLimit>
      <search:topic>
         <search:maxTreeDepth>2</search:maxTreeDepth>
         <search:maxChildrenPerNode>20</search:maxChildrenPerNode>
         <search:minDocsPerNode>2</search:minDocsPerNode>
      </search:topic>
      <search:metaData>
         <search:maxTreeDepth>3</search:maxTreeDepth>
         <search:maxChildrenPerNode>30</search:maxChildrenPerNode>
         <search:minDocsPerNode>3</search:minDocsPerNode>
      </search:metaData>
   </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 2-7.

<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="11.2.2.2.0"
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 2-137.

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 configurations related to the crawler:

```
<search:crawlerSettings>
   <search:numThreads>
   <search:numProcessors>
   <search:crawlDepth>
      <search:limit>
   <search:languageDetection>
   <search:defaultLanguage>
   <search:crawlTimeout>
   <search:maxDocumentSize>
   <search:charSetDetection>
   <search:defaultCharset>
   <search:preserveDocumentCache>
   <search:servicePipeline>
      <search:pipelineName>
   <search:verboseLogging>
   <search:logLanguage>
   <search:logLevel>
   <search:badTitles>
      <search:badTitle>
   <search:minCacheQueue>
   <search:maxCacheQueue>
   <search:fileWriteBufferSize>
   <search:idmUserCacheSize>
   <search:idmGroupCacheSize>
   <search:portalIndexContainerPage>
```

```
<search:portalSmartIncrCrawl>
<search:zipFilePackage>
<search:archiveFileTraverseDepth>
```

Element Descriptions

<search:crawlerSettings>

Contains the elements for configuring the crawler that are described in the following paragraphs.

<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 `true` to restrict crawling to the depth limit, or set to `false` 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 `true` to attempt to detect a language, or set to `false` 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, " Languages Supported by the Crawler".

<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:charSetDetection>
Controls whether to detect the character set automatically.

Attribute	Value
`enabled`	Set to `true` to detect the character set automatically, or set to `false` otherwise. Required.

<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:preserveDocumentCache>
Controls whether the cache is saved after indexing.

Attribute	Value
`enabled`	Set to `true` to preserve the cache, or set to `false` 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 `true` to enable the pipeline, or set to `false` 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 true to record all information, or set to false to record only summary information. Required.

<search:logLanguage>

Contains the language code for messages written to the log file. See Table 2–3, " Languages Supported by the Crawler".

<search:logLevel>

Contains the log level for the crawler. The following are the valid log levels:

Logging Level	Description
TRACE	Trace messages
DEBUG	Debug messages
INFO	Informational messages (Default)
WARN	Warning messages
ERROR	Error messages
FATAL	Fatal messages

<search:badTitles>

Contains one or more <search:badTitle> elements. This parameter can be set at the global level.

<search:badTitle>

Contains an exact character string for a document title that the crawler omits from the index. These bad titles are defined by default:

```
PowerPoint Presentation
Slide 1
```

<search:minCacheQueue>

Minimum size of the cache queue. The default size is 1MB.

<search:maxCacheQueue>

Maximum size of the cache queue. The default size is 10MB.

<search:fileWriteBufferSize>

Buffer size for writing files to disk.

<search:idmUserCacheSize>

Size of the user cache. This cache is used to avoid repeated lookups. The default size is 5000MB.

<search:idmGroupCacheSize>

Size of the group cache. This cache is used to avoid repeated lookups. The default size is 5000MB.

<search:portalIndexContainerPage>

Controls whether the portal container pages should be indexed, or they should be only used as *seeds* to crawl other portal items.

<search:portalSmartIncrCrawl>
Controls whether the portal container pages should be crawled incrementally, that is, only the portal container pages that were changed since the last re-crawl date should be crawled.

<search:zipFilePackage>
Specifies the Java package to use for processing zip files. The available options are `JDK` and `Apache`.

<search:archiveFileTraverseDepth>
Specifies the crawling depth for recursively traversing the nested archive files, such as, zip files.

Example

This XML document configures the crawler:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
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:charSetDetection enabled="true"/>
      <search:defaultCharSet>8859_1</search:defaultCharSet>
      <search:preserveDocumentCache enabled="true"/>
      <search:servicePipeline enabled="true">
         <search:pipelineName>Default pipeline</search:pipelineName>
      </search:servicePipeline>
      <search:verboseLogging enabled="true"/>
      <search:logLanguage>en-US</search:logLanguage>
      <search:logLevel>INFO</search:logLevel>
      <search:badTitles>
         <search:badTitle>PowerPoint Presentation</search:badTitle>
         <search:badTitle>Slide 1</search:badTitle>
      </search:badTitles>
      <search:minCacheQueue>1</search:minCacheQueue>
      <search:maxCacheQueue>10</search:maxCacheQueue>
      <search:fileWriteBufferSize>32K</search:fileWriteBufferSize>
      <search:idmUserCacheSize>5000</search:idmUserCacheSize>
      <search:idmGroupCacheSize>5000</search:idmGroupCacheSize>
      <search:portalIndexContainerPage>true</search:portalIndexContainerPage>
      <search:portalSmartIncrCrawl>true</search:portalSmartIncrCrawl>
      <search:zipFilePackage>JDK</search:zipFilePackage>
      <search:archiveFileTraverseDepth>3</search:archiveFileTraverseDepth>
   </search:crawlerSettings>
</search:config>
```

docServiceInstance

A document service instance is a Java class that implements the document service API. It accepts input from documents and performs an operation on it. For example, you could create a document service for auditing or to show custom metatags.

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 - Document Services - Create or Edit Document Service Instance

XML Description

The `<search:docServiceInstances>` element describes the document service instances:

```
<search:docServiceInstances>
   <search:docServiceInstance>
      <search:name>
      <search:instanceManagerName>
      <search:parameters>
         <search:parameter>
            <search:value>
            <search:description>
```

Element Descriptions

<search:docServiceInstances>
Describes all document service instances. It contains one or more `<search:docServiceInstance>` elements, each defining a document service instance.

<search:docServiceInstance>
Describes a document service instance. It contains these elements:

```
<search:name>
<search:instanceManagerName>
<search:parameters>
```

<search:name>
Contains the name of the document service instance.

<search:instanceManagerName>
Contains the name of the manager for the document service instance. (Read only)

<search:parameters>
Contains one or more `<search:parameter>` elements, each describing a parameter of the document service instance.

<search:parameter>
Describes a parameter. It contains these elements:

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

Attribute	Value
`name`	Name of the parameter. (Read only)

<search:value>
Contains the value of the parameter.

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

<search:description>
Contains a description of the parameter. (Read only)

Example

This XML document describes the default image service:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:docServiceInstances>
      <search:docServiceInstance>
         <search:name>Default image service instance</search:name>
         <search:instanceManagerName>
            Secure Enterprise Search Image Document Service
         </search:instanceManagerName>
         <search:parameters>
            <search:parameter name="attributes configuration file">
```

```
               <search:value>attr-config.xml</search:value>
              <search:description>EQG-12011:en-US:</search:description>
            </search:parameter>
         </search:parameters>
      </search:docServiceInstance>
   </search:docServiceInstances>
</search:config>
```

docServiceManager

A document service manager identifies the parameters for one or more document service instances.

Object Type

Creatable

Object Key

```
jarFilePath managerClassName
```

Object Key Command Syntax

```
--JAR_FILE=jarfile_name --MANAGER_CLASS=class
```

State Properties

None

Supported Operations

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

Administration GUI Page

Global Settings - Document Services - Service Managers

XML Description

The <search:docServiceManagers> element describes all document service managers:

```
<search:docServiceManagers>
   <search:docServiceManager>
      <search:managerClassName>
      <search:jarFilePath>
      <search:name>
      <search:description>
      <search:parameterInfos>
         <search:parameterInfo>
            <search:defaultValue>
            <search:encrypted>
            <search:description>
```

Element Descriptions

<search:docServiceManagers>
Describes all document service managers. It contains one or more <search:docServiceManager> elements, each defining a document service manager.

<search:docServiceManager>
Describes a document service manager. It contains these elements:

```
<search:managerClassName>
<search:jarFilePath>
<search:name>
<search:description>
<search:parameterInfo>
```

<search:managerClassName>
Contains the class name of the manager plug-in.

<search:jarFilePath>
Contains the qualified name of the jar file. Paths can be absolute or relative path to the *ses_home*/search/lib/plugins/doc directory.

<search:name>
Contains the name of the document service manager. (Read only)

<search:description>
Contains a description of the object. (Read only)

<search:parameterInfos>
Contains one or more `<search:parameterInfo>` elements, each describing a parameter of the document service manager. (Read only)

<search:parameterInfo>
Describes a parameter. (Read only)

This element contains these child elements:

```
<search:defaultValue>
<search:encrypted>
<search:description>
```

Attribute	Value
`name`	Name of the parameter. (Read only)

<search:defaultValue>
Contains the default value of the parameter. (Read only)

<search:encrypted>
Indicates whether the parameter represents a value that should be encrypted (Read only).

<search:description>
Description of the parameter.

Example

This XML document describes the Image Document Service Manager.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
      <search:docServiceManager>
         <search:managerClassName>
            oracle.search.plugin.doc.ordim.ImageDocumentServiceManager
         </search:managerClassName>
         <search:jarFilePath>ordim/ordimses.jar</search:jarFilePath>
         <search:name>ImageDocumentService</search:name>
         <search:description>
            document service that processes JPEG, GIF, TIFF, JPEG 2000 andDICOM
image metadata for search
         </search:description>
         <search:parameterInfos>
            <search:parameterInfo name="attributes configuration file">
               <search:defaultValue>attr-config.xml</search:defaultValue>
               <search:encrypted>false</search:encrypted>
               <search:description>
name of the configuration file that defined search attributes for image documents.
The file must exist at search/lib/plugins/doc/ordim/config.
               </search:description>
            </search:parameterInfo>
         </search:parameterInfos>
      </search:docServiceManager>
   </search:docServiceManagers>
</search:config>
```

docServicePipeline

A document service pipeline is a list of document service instances that are invoked in the order of the list. The same instance can be assigned to different pipelines, but it cannot be assigned twice in the same pipeline. You can have multiple pipeline definitions; for example, one pipeline could be used globally and another pipeline used for certain sources. An instance does not need to be in a pipeline.

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 - Document Services - Create or Edit Document Service Pipeline

XML Description

The `<search:docServicePipelines>` element describes the document service pipelines:

```
<search:docServicePipelines>
   <search:docServicePipeline>
      <search:name>
      <search:description>
      <search:assignedSources>
         <search:assignedSource>
      <search:serviceInstances>
         <search:serviceInstance>
```

Element Descriptions

<search:docServicePipelines>
Describes all document service pipelines. It contains one or more `<search:docServicePipeline>` elements, each defining a document service pipeline.

<search:docServicePipeline>
Describes a document service pipeline. It contains these elements:

```
<search:name>
<search:description>
<search:assignedSources>
<search:serviceInstances>
```

<search:name>
Contains the name of the document service pipeline.

<search:description>
Contains a description of the pipeline.

<search:assignedSources>
Contains one or more `<search:assignedSource>` element, each describing a source that the document service pipeline is assigned to. (Read only)

<search:assignedSource>
Contains the name of a source crawled using this pipeline. (Read only)

<search:serviceInstances>
Contains one or more `<search:serviceInstance>` elements, each describing an existing document service instance to be invoked by the document service pipeline.

<search:serviceInstance>
Contains the name of an existing document service instance to be invoked by the document service pipeline.

Example

This XML document describes a document service pipeline:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:docServicePipelines>
      <search:docServicePipeline>
         <search:name>My pipeline</search:name>
         <search:description>
            My document service pipeline
         </search:description>
         <search:assignedSources>
            <search:assignedSource>
               this_web_source
            </search:assignedSource>
            <search:assignedSource>
               that_web_source
            </search:assignedSource>
         </search:assignedSources>
         <search:serviceInstances>
            <search:serviceInstance>
               My web service instance
         </search:serviceInstance>
         </search:serviceInstances>
      </search:docServicePipeline>
   </search:docServicePipelines>
</search:config>
```

facetTree

Facets are a way of categorizing the search result data, so that the search results can be filtered based on various categories and sub-catgories. A facet tree is a hierarchy of categories and sub-catgories, where each category is called as a facet node, and can be used to narrow the number of matching documents.

A facet tree (facet name and facet node names) can be translated into different languages.

Object Type

Creatable

Object Key

FACETNAME

Object Key Command Syntax

```
--FACETNAME=name
```

State Properties

None

Supported Operations

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

Administration GUI Page

Global Settings - Facets

XML Description

The <search:facetTrees> element describes facet trees:

```
<search:facetTrees>
   <search:facetTree>
      <!-- Properties -->
      <search:facetName>
      <search:facetType>
      <search:mappedSearchAttribute>
      <search:pathDelimiter>
      <!-- Facet Name Translations -->
      <search:translations>
         <search:translation>
```

```
            <search:translatedValue>
        <!-- Facet Nodes -->
        <search:facetNodes>
           <search:facetNode>
              <search:nodeName>
              <search:matchExpression>
              <!-- Facet Node Name Translations -->
              <search:translations>
                 <search:translation>
                    <search:translatedValue>
```

Element Descriptions

<search:facetTrees>

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

<search:facetTree>

Describes a facet tree. It contains these elements:

```
<search:facetName>
<search:facetType>
<search:mappedSearchAttribute>
<search:translations>
<search:facetNodes>
```

<search:facetName>

Name of the facet. The maximum length is 2000 bytes in UTF-8. Required.

<search:facetType>

Data type of `<search:mappedSearchAttribute>`. Set to `STRING` only. Required.

<search:mappedSearchAttribute>

Name of the search attribute whose values are used as the facet values. The data type must be the same as `<search:facetType>`. The maximum length of a string facet is 2000 bytes in UTF-8 format.

<search:pathDelimiter>

Facet tree path delimiter, which is a slash (/) by default. The backslash (\) is the escape character, thus you must enter two backslashes (\\) to set the delimiter to a backslash.

<search:translations>

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

<search:translation>

Controls the translation language for the facet name. It contains `<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". Default is `en`, that is, English.

<search:translatedValue>

The translated value of the facet name in the specified language.

<search:facetNodes>
Describes the facet nodes for number and date data types. It contains one or more `<search:facetNode>` elements.

<search:facetNode>
Describes a facet node. It contains these elements:

```
<search:nodeName>
<search:matchExpression>
<search:translations>
```

<search:nodeName>
Name of the facet node. It can be a full path, starting from the root node. Any node in the path that does not exist is created automatically. The name must be unique within the parent node. The maximum length is 2000 bytes in UTF-8.

<search:matchExpression>
Provides an optional, conditional expression for number and date facets in the form:

```
type = [range | system], option operator expression, ...
```

> **Note:** Some of these characters have special significance in XML, so you must enter the entity references instead of the characters as element values:
>
> `&quot;` for `"` (quotation marks)
> `&amp;` for `&` (ampersand)

The expression can be one of these types:

- `range`: Uses the `beginValue` and `endValue` parameters to identify a range of values. You can specify one or both of these parameters. Use the `include` parameter to identify whether the range value includes or excludes the beginning value and the end value of the range. Use the `interval` parameter to indicate the time interval in days, months, or years.

  `beginValue [ = ] [`*`expression`*`]`: Identifies the beginning of the range. Optional if `endValue` is specified

  `endValue [ = ] [`*`expression`*`]`: Identifies the end of the range. Optional if `beginValue` is specified.

  `include [ = ] [begin | end | both | none]`: Identifies whether the range value includes or excludes the beginning value and the end value of the range. Specify `begin` to include the beginning value, specify `end` to include the end value, specify `both` to include beginning value as well as end value of the range, and specify `none` to exclude beginning value as well as end value of the range. If the `include` parameter for the range expression type is not specified, then the default processing is same as that of `begin`, that is, the beginning value of the range is included.

  `interval [ = ] [DAY | MONTH | YEAR]`: This range parameter can be used only for the date type facets. It indicates the time interval in days, months, or years.

- `system`: This expression type can be used only with the date type facets. Uses the `value` parameter to specify a predefined date range using the keywords described as follows:

  ```
  value = [Today | Yesterday | This Week | This Month | This Year | Before This
  ```

```
    Year]
```

<search:translations>
Contains one or more `<search:translation>` elements.

<search:translation>
Controls the translation language for the facet node name. It contains `<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". Default is `en`, that is, English.

<search:translatedValue>
The translated value of the facet node name in the specified language.

Example

This XML document describes three facet trees. Books is a string facet tree. Price is a number facet tree with three nodes: Under $20, Under $35, and $35 and up. Published Date is a date facet tree with three nodes also: New Releases, Recent Titles, and Timeless Treasures.

```
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
<search:facetTrees>
   <search:facetTree>
      <search:facetName>Books</search:facetName>
      <search:facetType>STRING</search:facetType>
      <search:mappedSearchAttribute>Title</search:mappedSearchAttribute>
   </search:facetTree>

   <search:facetTree>
      <search:facetName>Price</search:facetName>
      <search:facetType>NUMBER</search:facetType>
      <search:mappedSearchAttribute>Price</search:mappedSearchAttribute>
      <search:facetNodes>
         <search:facetNode>
       <search:nodeName>Under $20</search:nodeName>
       <search:matchExpression>
          type=range, endValue=20, include=none
       </search:matchExpression>
    </search:facetNode>
    <search:facetNode>
       <search:nodeName>Under $35</search:nodeName>
       <search:matchExpression>
          type=range, endValue=35, beginValue=20,include=begin
       </search:matchExpression>
    </search:facetNode>
         <search:facetNode>
       <search:nodeName>$35 and up</search:nodeName>
       <search:matchExpression>
          type=range, beginValue=35, include=begin
       </search:matchExpression>
    </search:facetNode>
      </search:facetNodes>
   </search:facetTree>
   <search:facetTree>
      <search:facetName>Published Date</search:facetName>
      <search:facetType>DATE</search:facetType>
      <search:mappedSearchAttribute>Year</search:mappedSearchAttribute>
      <search:facetNodes>
         <search:facetNode>
       <search:nodeName>New Releases</search:nodeName>
       <search:matchExpression>
          type=system, value=This Year
       </search:matchExpression>
    </search:facetNode>
    <search:facetNode>
       <search:nodeName>Recent Titles</search:nodeName>
       <search:matchExpression>
          type=system, value=Before This Year
       </search:matchExpression>
    </search:facetNode>
    <search:facetNode>
       <search:nodeName>Timeless Treasures</search:nodeName>
       <search:matchExpression>
          type=range, endValue=today - 2, interval=YEAR, include=end
       </search:matchExpression>
    </search:facetNode>
      </search:facetNodes>
   </search:facetTree>
</search:facetTrees>
</search:config>
```

Example Contents of <search:matchExpression>

Following are the examples of the content of the `<search:matchExpression>` element.

Example 1 Number Data Type

For range `[10 - 20]` where 10 is inclusive and 20 is exclusive:

```
type=range, beginValue=10, endValue=20
```

or

```
type=range, beginValue=10, endValue=20, include=begin
```

For range `[10 - 20]` where both 10 and 20 are inclusive:

```
type=range, beginValue=10, endValue=20, include=both
```

For range `[* - 100]` for any number less than 100 (exclusive):

```
type=range, endValue=100, include=none
```

For range `[100 - *]` for any number greater than or equal to 100:

```
type=range, beginValue=100, include=begin
```

For range [100 - 100] where every element in the range has a value of 100:

```
type=range, beginValue=100
```

Example 2 Date Data Type: Absolute Value

Specify absolute values for dates using the format *mm/dd/yyyy*.

For range [year 2001- year 2011] for years 2001 and 2011 inclusive:

```
type=range, beginValue=01/01/2001, endValue=12/31/2011, include=both
```

For range [* - year 2010] for any date before or in year 2010:

```
type=range, endValue=12/31/2010, include=end
```

For range [year 2000 - *] for any date in or after year 2001:

```
type=range, beginValue=01/01/2001, include=begin
```

For range [year 2001] for any date that matches year 2001:

```
type=range, beginValue=01/01/2001, endValue=01/01/2002
```

For range [01/01/2011 - 01/01/2011] for any date that exactly matches the date January 1, 2011:

```
type=range, beginValue=01/01/2011, endValue=01/01/2011, include=both
```

Example 3 Date Data Type: Relative Value

Last three years including this year up to today:

```
type=range, beginValue=today - 3, endValue=today, include=end, interval=YEAR
```

Last seven days including today:

```
type=range, beginValue=today - 7, endValue=today, include=end, interval=DAY
```

Last six months including this month up to today:

```
type=range, beginValue=today - 6, endValue=today, include=end, interval=MONTH
```

Example 4 Date Data Type: Predefined Constants

Today:

```
type = system, value = Today
```

This year:

```
type = system, value = This Year
```

globalBoundaryRules

The default boundary rules specified in this object are copied to new sources that are created with no other boundary rules.

Boundary rules restrict the crawler to those URLs that match the specified rules. Exclusion rules override inclusion rules. The order in which the rules are listed has no impact.

For file sources with no boundary rules, crawling is limited to the underlying file system access privileges. Files accessible from the specified seed file URL are crawled to the default crawling depth.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

None

XML Description

The `<search:globalBoundaryRules>` element describes the rules limiting the scope of the crawler. It contains these elements:

```
<search:globalBoundaryRules>
   <search:boundaryRules>
      <search:boundaryRule>
         <search:ruleType>
         <search:ruleOperation>
         <search:rulePattern>
```

Element Descriptions

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

<search:boundaryRules>
Contains one or more `<search:boundaryRule>` elements.

<search:boundaryRule>
Describes a boundary rule. It contains these child elements:

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

<search:ruleType>
Type of URL boundary rule:

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

<search:ruleOperation>
Matching operation for a search rule pattern:

- 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 matches 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 (?) matches zero or one occurrence of the character that it follows.
- Asterisk (*) matches zero or more occurrences of the pattern that it follows. 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.

Files with the following filename extensions are excluded by the default boundary rule patterns:

- **Image**: bmp, png, tif
- **Audio**: wav, wma, mp3
- **Video**: avi, wmv, mpeg, mpg
- **Binary**: bin, cab, dll, dmp, ear, exe, iso, jar, scm, so, tar, war, wmv

Example

This XML document defines the default global boundary rules:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:globalBoundaryRules>
      <search:boundaryRules>
         <search:boundaryRule>
            <search:ruleType>EXCLUSION</search:ruleType>
            <search:ruleOperation>REGEX</search:ruleOperation>
            <search:rulePattern>
(?i:(?:\.jar)|(?:\.bmp)|(?:\.war)|(?:\.ear)|(?:\.mpg)|(?:\.wmv)|(?:\.mpeg)|(?:\.sc
m)|(?:\.iso)|(?:\.dmp)|(?:\.dll)|(?:\.cab)|(?:\.so)|(?:\.avi)|(?:\.wav)|(?:\.mp3)|
(?:\.wma)|(?:\.bin)|(?:\.exe)|(?:\.iso)|(?:\.tar)|(?:\.png))$
            </search:rulePattern>
         </search:boundaryRule>
         <search:boundaryRule>
            <search:ruleType>EXCLUSION</search:ruleType>
            <search:ruleOperation>REGEX</search:ruleOperation>
            <search:rulePattern>\?.*(.*\+)\1{3}</search:rulePattern>
         </search:boundaryRule>
      </search:boundaryRules>
   </search:globalBoundaryRules>
</search:config>
```

globalDocumentTypes

This object defines the default document types for each new source.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

None

XML Description

The <search:documentTypes> element describes the default document types:

```
<search:globalDocumentTypes>
   <search:documentTypes>
      <search:documentType>
         <search:mimeType>
```

Element Descriptions

<search:globalDocumentTypes>
Contains one or more <search:documentTypes> elements.

<search:documentTypes>
Contains one or more <search:documentType> elements.

Attribute	Value
processAll	Set to true to process all the MIME types by default, or set to false otherwise.

<search:documentType>
Contains a <search:mimeType> element.

<search:mimeType>
Contains a supported MIME type, as described in Table 2–1, " Document Formats Supported by Oracle SES". These MIME types are defined by default:

```
application/msword
application/pdf
application/x-msexcel
application/x-mspowerpoint
text/html
text/plain
```

Example

This XML document describes the default global document types:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:globalDocumentTypes>
      <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>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:documentTypes>
   </search:globalDocumentTypes>
</search:config>
```

To process all the supported MIME types by default, specify `processAll=true` for `<search:documentTypes>` element:

```
<?xml version="1.0" encoding="UTF-8"?>
   <search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:globalDocumentTypes>
      <search:documentTypes processAll="true"/>
   <search:globalDocumentTypes/>
</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 3-7.

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 *ses_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="11.2.2.2.0"
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="11.2.2.2.0"
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
stop
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="11.2.2.2.0"
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>
```

indexProfile

An index profile is a group of index settings that can be used by multiple data sources. All newly created sources use the default index profile, which you can configure in the crawlerSettings object.

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

The <search:indexProfiles> element describes all the index profiles:

```
<search:indexProfiles>
   <search:indexProfile>
      <search:name>
      <search:description>
      <search:tokenization>
         <search:languageBasedTokenization>
         <search:userDefinedTokenization>
            <search:mappingRules>
               <search:mappingRule>
                  <search:rule>
                  <search:lexerName>
            <search:defaultMappingRule>
```

```
        <search:lexerName>
```

Element Descriptions

<search:indexProfiles>
Contains one or more `<search:indexProfile>` elements.

<search:indexProfile>
Describes an index profile. It contains these elements:

```
<search:name>
<search:description>
<search:tokenization>
```

<search:name>
Contains the name of the index profile. Required.

<search:description>
Contains a description of the index profile.

<search:tokenization>
Identifies the type of tokenization used by the index profile. It contains one of these elements:

```
<search:userDefinedTokenization>
<search:languageBasedTokenization
```

<search:languageBasedTokenization>
Tokenization is performed using the language mapping rules defined in a languageBasedTokenization object. Default.

<search:userDefinedTokenization>
Tokenization is performed using a set of prioritized mapping rules that you define in this element. It contains a `<search:mappingRules>` element.

<search:mappingRules>
Contains one or more `<search:mappingRule>` elements.

<search:mappingRule>
Associates a rule with lexer and identifies the priority of the rule. It contains these elements:

```
<search:rule>
<search:lexerName>
```

`priority`	A positive integer that identifies the priority of the rule. Each rule must have a unique priority number. Rules are evaluated in numeric order: If the first rule does not match the document, then the second rule is evaluated, and so forth.

<search:rule>
A text string in the form *`attribute=value`*:

`attribute` is the name of a String document attribute.

`value` is a value of the attribute. It can contain an asterisk (*) as a wildcard. To use an asterisk or a backslash (\) as literal values, precede them with a backslash as an escape (* or \\).

Both parts of the string are case-insensitive.

<search:lexerName>
Contains the name of a lexer object.

<search:defaultMappingRule>
Identifies the lexer for a document that does not match any of the other mapping rules. It contains a <search:lexerName> element. Required.

<search:lexerName>
Contains the name of the default lexer object.

Example

This XML document describes the default index profile, which uses language-based tokenization and a custom index profile, which defines the tokenization rules.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <search:indexProfiles>
    <search:indexProfile>
      <search:name>Oracle Default Index Profile</search:name>
      <search:description>
        Oracle Secure Enterprise Search Default index profile.
      </search:description>
      <search:tokenization>
        <search:languageBasedTokenization/>
      </search:tokenization>
    </search:indexProfile>
    <search:indexProfile>
      <search:name>This Index Profile</search:name>
      <search:description>Alternate index profile</search:description>
      <search:tokenization>
        <search:userDefinedTokenization>
          <search:mappingRules>
            <search:mappingRule priority="1">
              <search:rule>Language=en</search:rule>
              <search:lexerName>OracleDefaultLanguageLexer</search:lexerName>
            </search:mappingRule>
            <search:mappingRule priority="2">
              <search:rule>Language=de</search:rule>
              <search:lexerName>OracleDefaultGermanLexer</search:lexerName>
            </search:mappingRule>
            <search:mappingRule priority="3">
              <search:rule>Mimetype=text/html</search:rule>
              <search:lexerName>OracleDefaultLanguageLexer</search:lexerName>
            </search:mappingRule>
          </search:mappingRules>
          <search:defaultMappingRule>
            <search:lexerName>OracleDefaultLanguageLexer</search:lexerName>
          </search:defaultMappingRule>
        </search:userDefinedTokenization>
      </search:tokenization>
    </search:indexProfile>
  </search:indexProfiles>
</search:config>
```

languageBasedTokenization

Language-based tokenization associates each document language with a lexer.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

None

XML Description

The <search:languageBasedTokenization> element describes language-based tokenization:

```
<search:languageBasedTokenization>
   <search:languageMappingRules>
      <search:languageMappingRule>
         <search:language>
         <search:lexerName>
      <search:defaultLanguageMappingRule>
         <search:lexerName>
```

Element Descriptions

<search:languageBasedTokenization>
Contains a <search:languageMappingRules> element.

<search:languageMappingRules>
Contains one or more <search:languageMappingRule> elements.

<search:languageMappingRule>
Identifies the lexer used to tokenize a document language.

<search:language>
Contains a two-letter language code from Table 2–3. A language can be mapped only once.

<search:lexerName>
Contains the name of the lexer to use to tokenize documents in the specified language.

<search:defaultLanguageMappingRule>
Identifies the lexer to use for document languages without a language mapping rule. It contains a <search:lexerName> element. (Required)

<search:lexerName>
Contains the name of the default lexer to use to tokenize documents in the specified language.

Example

This XML document describes the mapping rules for language-based tokenization:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:languageBasedTokenization>
      <search:languageMappingRules>
         <search:languageMappingRule>
            <search:language>de</search:language>
            <search:lexerName>OracleDefaultGermanLexer</search:lexerName>
         </search:languageMappingRule>
         <search:languageMappingRule>
            <search:language>ja</search:language>
            <search:lexerName>OracleDefaultJapaneseLexer</search:lexerName>
         </search:languageMappingRule>
         <search:languageMappingRule>
            <search:language>ko</search:language>
            <search:lexerName>OracleDefaultKoreanLexer</search:lexerName>
         </search:languageMappingRule>
         <search:languageMappingRule>
            <search:language>zh</search:language>
            <search:lexerName>OracleDefaultChineseLexer</search:lexerName>
         </search:languageMappingRule>
      </search:languageMappingRules>
      <search:defaultLanguageMappingRule>
         <search:lexerName>OracleDefaultLanguageLexer</search:lexerName>
      </search:defaultLanguageMappingRule>
   </search:languageBasedTokenization>
</search:config>
```

lexer

Lexers convert a sequence of characters into tokens. Different languages and different data sets require different tokenization rules. Oracle SES uses lexers to tokenize documents for indexing and to tokenize queries. For indexing, the crawler identifies the document language and determines the correct lexer to tokenize each document. For queries, Oracle SES uses a single lexer based on the user's browser language.

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
```

* A lexer cannot be deleted when a languageBasedTokenization object is using it.

Administration GUI Page

None

XML Description

The <search:lexers> element describes the language lexers:

```
<search:lexers>
   <search:lexer>
      <search:name>
      <search:type>
      <search:description>
      <search:parameters>
         <search:parameter>
         <search:value>
```

Element Descriptions

<search:lexers>
Contains one or more `<search:lexer>` elements.

<search:lexer>
Describes a lexer. It contains these elements:

```
<search:name>
<search:type>
<search:description>
<search:parameters>
```

<search:name>
Contains a case-insensitive name that uniquely identifies the lexer. The name cannot contain spaces, be more than 26 bytes in UTF-8, or begin with the string `Oracle`.

<search:type>
Contains a supported lexer type. You cannot change the type after the lexer is created.

- BASIC_LEXER
- CHINESE_LEXER
- CHINESE_VGRAM_LEXER
- JAPANESE_LEXER
- JAPANESE_VGRAM_LEXER
- KOREAN_MORPH_LEXER

Oracle SES uses a subset of Oracle Text lexers. For more information about these lexers, refer to the *Oracle Text Reference*.

<search:description>
Contains a description of the lexer.

<search:parameters>
Contains one or more `<search:parameter>` elements.

<search:parameter>
Contains the name of a supported attribute for the lexer type, which are described in the following topics. All attribute names are case-insensitive.

- BASIC_LEXER
- CHINESE_LEXER
- CHINESE_VGRAM_LEXER
- JAPANESE_LEXER
- JAPANESE_VGRAM_LEXER
- KOREAN_MORPH_LEXER

<search:value>
Contains the value of the attribute. All attribute values are case-insensitive.

To specify white space in a parameter value, enter the string `SPACE`.

BASIC_LEXER

The BASIC_LEXER type identifies tokens for English and all other supported whitespace-delimited languages. You can use it with any database character set.

BASIC_LEXER Attributes

continuation
One or more characters that indicate a word continues on the next line and should be indexed as a single token. The most common continuation characters are hyphen '-' and backslash '\'.

numgroup
A single character that, when it appears in a string of digits, indicates that the digits are groupings within a larger single unit. For example, comma ',' might be defined as a `numgroup` character because it often indicates a grouping of thousands when it appears in a string of digits.

The globalization support initialization parameters for the database determine the default value.

numjoin
One or more characters that, when they appear in a string of digits, indicates that the string should be indexed as a single unit or word. For example, period '.' might be defined as `numjoin` characters because it often serves as a decimal point when it appears in a string of digits.

The globalization support initialization parameters for the database determine the default value.

printjoins
One or more nonalphanumeric characters that, when they appear anywhere in a word, are processed as alphanumeric and included with the token in the index. This includes `printjoins` characters that occur consecutively.

For example, if the hyphen '-' and underscore '_' are defined as `printjoins` characters, then terms such as pseudo-intellectual and _file_ are stored in the index as `pseudo-intellectual` and `_file_`.

`Printjoins` differ from `endjoins` and `startjoins` in that position does not matter. For example, *$35* is indexed as one token if *$* is a `startjoin` or a `printjoin`, but as two tokens if it is defined as an `endjoin`.

If a `printjoins` character is also defined as a `punctuations` character, it is processed as a `printjoins` character only if the character immediately following it is a standard alphanumeric character, or it has been defined as a `printjoins` or `skipjoins` character.

punctuations
One or more nonalphanumeric characters that, when they appear at the end of a word, indicate the end of a sentence. The defaults are period '.', question mark '?', and exclamation point '!'.

Characters that are defined as `punctuations` are removed from a token before indexing. However, if a `punctuations` character is also defined as a `printjoins` character, then the character is removed only when it is the last character in the token.

For example, if the period (.) is defined as both a `printjoins` and a `punctuations` character, then the following transformations take place during indexing and querying as well:

Token	Indexed Token
.doc	.doc
dog.doc	dog.doc
dog..doc	dog..doc
dog.	dog
dog...	dog..

BASIC_LEXER use punctuations characters with newline and whitespace characters to determine sentence and paragraph delimiters for sentence/paragraph searching.

skipjoins

One or more nonalphanumeric characters that, when they appear within a word, identify the word as a single token; however, the characters are not stored with the token in the index.

For example, if the hyphen '-' is defined as a skipjoins character, then the word pseudo-intellectual is stored in the index as pseudointellectual.

Printjoins and skipjoins are mutually exclusive. The same characters cannot be specified for both attributes.

startjoins

One or more nonalphanumeric characters that, when encountered as the first character in a token, identify the start of the token. The character and any trailing startjoins characters are included in the index entry for the token. In addition, the first startjoins character in a string of startjoins characters implicitly ends the previous token.

The following rules apply:

- The specified characters cannot occur in any of the other attributes for BASIC_LEXER.
- The characters can occur only at the beginning tokens.

endjoins

One or more nonalphanumeric characters that, when encountered as the last character in a token, identify the end of the token. The character and any trailing startjoins characters are included in the Text index entry for the token.

The following rules apply:

- The characters cannot occur in any of the other attributes for BASIC_LEXER.
- The characters can occur only at the end of tokens.

whitespace

One or more characters that are treated as blank spaces between tokens. BASIC_LEXER uses whitespace characters with punctuations and newline characters to identify character strings that serve as sentence delimiters for sentence and paragraph searching.

The predefined default values for whitespace are space and tab. These values cannot be changed. Specifying characters as whitespace characters adds to these defaults.

newline

Characters that indicate the end of a line of text. BASIC_LEXER uses newline characters with punctuation and whitespace characters to identify character strings that serve as paragraph delimiters for sentence and paragraph searching.

The only valid values for newline are NEWLINE and CARRIAGE_RETURN (for carriage returns). The default is NEWLINE.

base_letter

YES to convert characters that have diacritical marks (umlauts, cedillas, acute accents, and so on) to their base form before being stored in the index. The default is NO, which disables base-letter conversion.

base_letter_type

The transformation table for base-letter transformations:

GENERIC uses one transformation table for all languages for base-letter transformation (default).

SPECIFIC uses different transformation tables for different languages.

override_base_letter

TRUE prevents unexpected results from serial transformations when base_letter is enabled at the same time as alternate_spelling. Default is FALSE.

composite

DUTCH and GERMAN enable composite word indexing for the specified language. DEFAULT disables composite word indexing (default).

Words that are usually one entry in a German dictionary are not split into composite stems, while words that are not dictionary entries are split into composite stems.

alternate_spelling

DANISH, GERMAN, and SWEDISH enable alternate spelling in the specified language. Users can then query a word in any of its alternate forms. NONE disables alternate spelling in all languages.

Alternate spelling is typically off by default, but may be on for some German-, Danish-, and Swedish-language installations.

new_german_spelling

YES returns both traditional and reformed (new) spellings of German words. NO matches words only as they are entered in the query (default).

CHINESE_LEXER

The CHINESE_LEXER type identifies tokens in traditional and simplified Chinese text. It generates a smaller index and supports better query response time than the CHINESE_VGRAM_LEXER type, but indexing takes longer.

You can use this lexer if your database uses a Chinese or Unicode character sets supported by Oracle. See the *Oracle Database Globalization Support Guide*.

The CHINESE_LEXER type has no attributes.

CHINESE_VGRAM_LEXER

The CHINESE_VGRAM_LEXER type identifies tokens in Chinese text. Indexing is quicker than the CHINESE_LEXER type, but the index is larger and querying is slower.

You can use this lexer if your database uses one of these character sets:

AL32UTF8
UTF8
ZHS16CGB231280
ZHS16GBK
ZHS32GB18030
ZHT32EUC
ZHT16BIG5
ZHT32TRIS
ZHT16HKSCS
ZHT16MSWIN950

The CHINESE_VGRAM_LEXER type has no attributes.

JAPANESE_LEXER

The JAPANESE_LEXER type identifies tokens in Japanese. It generates a smaller index and supports better query response time than the JAPANESE_VGRAM_LEXER type, but indexing takes longer.

You can use this lexer if your database uses one of these character sets:

AL32UTF8
UTF8
JA16SJIS
JA16EUC
JA16EUCTILDE
JA16EUCYEN
JA16SJISTILDE
JA16SJISYEN

JAPANESE_LEXER Attributes

delimiter
Specify `NONE` or `ALL` to ignore certain Japanese blank characters, such as a full-width slash or a full-width middle dot. Default is `NONE`.

JAPANESE_VGRAM_LEXER

The JAPANESE_VGRAM_LEXER type identifies tokens in Japanese. Indexing is quicker than the JAPANESE_LEXER type, but the index is larger and querying is slower.

You can use this lexer if the database uses one of these character sets:

AL32UTF8
UTF8
JA16SJIS
JA16EUC
JA16EUCTILDE
JA16EUCYEN
JA16SJISTILDE
JA16SJISYEN

JAPANESE_VGRAM_LEXER Attributes

delimiter
Specify `NONE` or `ALL` to ignore certain Japanese blank characters, such as a full-width slash or a full-width middle dot. Default is `NONE`.

KOREAN_MORPH_LEXER

The KOREAN_MORPH_LEXER type identifies tokens in Korean text.

You can use this lexer if the database uses one of these character sets:

AL32UTF8
UTF8
KO16KSC5601
KO16MSWIN949

KOREAN_MORPH_LEXER Attributes

verb_adjective
`TRUE` to index verbs, adjectives, and adverbs, or `FALSE` to skip them (default).

one_char_word
`TRUE` to index one syllable tokens, or `FALSE` to skip them (default).

number
`TRUE` to index numbers, or `FALSE` to skip them (default).

composite
The indexing style of composite nouns:

`COMPOSITE_ONLY` indexes only composite nouns.

`NGRAM` indexes all noun components of a composite noun.

`COMPONENT_WORD` indexes single noun components of composite nouns and the composite noun itself (default).

morpheme
`TRUE` for morphological analysis (default), or `FALSE` to create tokens from words that are delimited, for example, by white space.

to_upper
`TRUE` to convert English to uppercase (default), or `FALSE` to retain mixed case.

hanja
`TRUE` to index hanja characters, or `FALSE` to convert hanja characters to hangul characters (default).

long_word
`TRUE` to index long words that have more than 16 syllables in Korean, or `FALSE` to skip them (default).

japanese
`TRUE` to index Japanese characters in Unicode (only in the 2-byte area), or `FALSE` to skip them (default).

english
`TRUE` to index alphanumeric strings (default), or `FALSE` to skip them.

Example

This XML document describes the default lexer for Oracle SES:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:lexers>
      <search:lexer>
         <search:name>OracleDefaultLanguageLexer</search:name>
         <search:type>BASIC_LEXER</search:type>
         <search:description>Oracle Secure Enterprise Search default lexer for all
languages except Chinese, German, Japanese and Korean.</search:description>
         <search:parameters>
            <search:parameter name="BASE_LETTER">
               <search:value>YES</search:value>
            </search:parameter>
         </search:parameters>
      </search:lexer>
   </search:lexers>
</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:** "Parallel Querying and Index Partitioning" in *Oracle Secure Enterprise Search Administrator's Guide*

Object Type

Universal

State Properties

Property	Value
`status`	`ACTIVE` `INACTIVE`

Supported Operations

```
activate
export
getState
update
```

Administration GUI Page

None

XML Description

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

```
<search:partitionConfig>
   <search:partitionAttrs>
      <search:partitionAttr>
         <search:name>
   <search:partitionRules>
      <search:partitionRule>
         <search:partitionValue>
         <search:valueType>
         <search:ruleType>
         <search:ruleSetting>
         <search:storageArea>
```

Element Descriptions

<search:partitionConfig>

Describes the partition configuration rules. It contains these elements:

```
<search:partitionAttrs>
 <search:partitionRules>
```

<search:partitionAttrs>
Contains a `<search:partitionAttr>` element.

<search:partitionAttr>
Describes an attribute on which partitioning is based. It contains a `<search:name>` element.

<search:name>
Contains the name of any String-type search attribute, or one of the following system-defined values:

- `EQ_SOURCE_NAME`: Data source name.
- `EQ_SOURCE_TYPE`: Data source type. To know all the data source types, export the sourceType object.

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

<search:partitionRule>
Describes a partition rule for the expected values of the partitioning attribute. It contains these elements:

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

<search:partitionValue>
Contains an expected value of the partitioning attribute or one of these values:

- `EQ_DEFAULT`: Identifies the partition rule when no partition attribute is defined.
- `EQ_OTHER`: Identifies the partition rule when none of the other defined values of `<search:partitionValue>` match the attribute value of the document.
- `EQ_OWNER`: Identifies the partition rule with the document owner information, such as, e-mail ID and global user ID.

<search:valueType>
Contains the type of partition value. Specify `ATTR` if it is an attribute value or specify `META` if it is a system-defined value, such as, `EQ_OTHER` or `EQ_DEFAULT`. 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.
- `BUCKETING`: The number of partitions are created based on the bucket size specified. While crawling, if the number of distinct partition attribute values extend beyond the specified bucket size, then the new partitions are created according to the bucket size specified. Only the `EQ_OWNER` and `EQ_OTHER` partition values can have `BUCKETING` rule type.

<search:ruleSetting>

Contains the rule setting for the BUCKETING rule type. It defines the *bucket size* for each partition. The bucket size denotes the maximum number of unmapped values for each partition. The format for specifying rule setting is:

```
bucket_size/storage_area_list
```

where,

bucket_size is the maximum number of unmapped values for each partition.

storage_area_list is the comma-delimited list of storage areas associated with a partition.

For example, the following rule setting for BUCKETING rule type specifies the bucket size of 10 for the two storage areas SA1 and SA2.

```
10/SA1,SA2
```

This rule will create the first partition (bucket) in SA1, the second partition in SA2, the third partition in SA1, the fourth partition in SA2, and so on. Thus, the partitions will be created in sequence of the list of storage areas defined. A new rule of BUCKET type will be automatically created for any new partition value encountered.

<search:storageArea>

For a VALUE rule, specify the name of a single storageArea object.

For a HASH rule, specify a comma-delimited list of storageArea objects used by this partition rule. Repeat the name of a storageArea object to create multiple partitions within a single tablespace. Remember that a storageArea object is a tablespace in Oracle Database that is registered for use with Oracle SES.

For example, this list creates one partition in each tablespace:

```
SA1, SA2, SA3
```

The next list creates three partitions in SA1 and two partitions in SA2:

```
SA1, SA1, SA1, SA2, SA2
```

Examples

This XML document describes partitioning of the document index across six storage areas named SA1 to SA6:

```
<search:config productVersion="11.2.2.2.0"
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>SA1, SA2, SA3, SA4, SA5, SA6</search:storageArea>
         </search:partitionRule>
      </search:partitionRules>
   </search:partitionConfig>
</search:config>
```

This example creates a partitioning rule based on the Language attribute. Documents with value of en (English) or ja (Japanese) for the Language attribute are indexed in the SA1 storage area. All the other documents are hashed into the SA2 and SA3 storage areas.

```
<search:config productVersion="11.2.2.2.0"
```

```
xmlns:search="http://xmlns.oracle.com/search">
   <search:partitionConfig>
      <search:partitionAttrs>
         <search:partitionAttr>
            <search:name>Language</search:name>
         </search:partitionAttr>
      </search:partitionAttrs>
      <search:partitionRules>
         <search:partitionRule>
            <search:partitionValue>en,ja</search:partitionValue>
            <search:valueType>ATTR</search:valueType>
            <search:ruleType>VALUE</search:ruleType>
            <search:storageArea>SA1</search:storageArea>
         </search:partitionRule>
         <search:partitionRule>
            <search:partitionValue>EQ_DEFAULT</search:partitionValue>
            <search:valueType>META</search:valueType>
            <search:ruleType>HASH</search:ruleType>
            <search:ruleSetting></search:ruleSetting>
            <search:storageArea>SA2,SA3</search:storageArea>
         </search:partitionRule>
      </search:partitionRules>
   </search:partitionConfig>
</search:config>
```

The next example stores the document index from the `Doc Library` source in SA1, from `My Web Site` source in SA2, and from all the other sources in SA3:

```
<search:config xmlns:search="http://xmlns.oracle.com/search"
productVersion="11.2.2.2.0">
   <search:partitionConfig>
     <search:partitionAttrs>
  <search:partitionAttr>
     <search:name>EQ_SOURCE_NAME</search:name>
  </search:partitionAttr>
</search:partitionAttrs>
      <search:partitionRules>
         <search:partitionRule>
            <search:partitionValue>Doc Library</search:partitionValue>
            <search:valueType>ATTR</search:valueType>
            <search:ruleType>VALUE</search:ruleType>
            <search:storageArea>SA1</search:storageArea>
         </search:partitionRule>
         <search:partitionRule>
            <search:partitionValue>My Web Site</search:partitionValue>
            <search:valueType>ATTR</search:valueType>
            <search:ruleType>VALUE</search:ruleType>
            <search:storageArea>SA2</search:storageArea>
         </search:partitionRule>
         <search:partitionRule>
            <search:partitionValue>EQ_OTHER</search:partitionValue>
            <search:valueType>META</search:valueType>
            <search:ruleType>VALUE</search:ruleType>
            <search:storageArea>SA3</search:storageArea>
         </search:partitionRule>
      </search:partitionRules>
   </search:partitionConfig>
</search:config>
```

The next example creates a partition rule of `BUCKETING` type with the bucket size of 10 for the two storage areas SA1 and SA2:

```
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:partitionConfig>
      <search:partitionRules>
         <search:partitionRule>
            <search:partitionValue>EQ_OWNER</search:partitionValue>
            <search:valueType>META</search:valueType>
            <search:ruleType>BUCKETING</search:ruleType>
            <search:ruleSetting>10/SA1,SA2</search:ruleSetting>
            <search:storageArea></search:storageArea>
         </search:partitionRule>
      </search:partitionRules>
   </search:partitionConfig>
</search:config>
```

proxy

The `proxy` object defines the HTTP proxy server settings.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

Global Settings - Proxy Settings

XML Description

The `<search:proxy>` element describes the HTTP proxy server parameters:

```
<search:proxy>
    <search:server>
    <search:port>
    <search:proxyExceptions>
        <search:proxyException>
```

Element Descriptions

<search:proxy>
Describes HTTP proxy server parameters. It contains these elements:

```
<search:server>
<search:port>
<search:proxyExceptions>
```

<search:server>
URL of HTTP proxy server.

<search:port>
Port number of HTTP proxy server.

<search:proxyExceptions>
Contains one or more `<search:proxyException>` elements. It contains a list of domain names that should not go through HTTP proxy server.

<search:proxyException>
A proxy exception, that is, a domain name that should not go through HTTP proxy server.

Example

This XML document contains HTTP proxy server configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:proxy>
      <search:server>www-proxy.us.xyz.com</search:server>
      <search:port>80</search:port>
      <search:proxyExceptions>
         <search:proxyException>*.us.example.com</search:proxyException>
         <search:proxyException>www.abc.com</search:proxyException>
      </search:proxyExceptions>
   </search:proxy>
</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="11.2.2.2.0"
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 search 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:lastCrawlDatesMergeRange>
  <search:searchTimeout>
  <search:enableWildcardQueries>
  <search:displayUrls>
     <search:tableDisplayUrl>
     <search:fileDisplayUrl>
     <search:mailingListDisplayUrl>
     <search:emailDisplayUrl>
  <<search:relevancyBoosting>
  <search:spellingCorrection>
     <search:useLanguageDictionary>
     <search:useIndexedDocsAndQueryLog>
  <search:hitCount>
     <search:countMethod>
     <search:maxExactHitCount>
  <search:queryStatistics>
     <search:loggingPeriod>
  <search:urlSubmission>
     <search:sourceName>
     <search:checkUrlBoundaryRules>
  <search:federation>
     <search:timeout>
     <search:minNumThreads>
     <search:maxNumThreads>
  <search:queryTimeAuthorization>
     <search:timeout>
     <search:minNumThreads>
     <search:maxNumThreads>
     <search:logFilteredDocs>
  <search:secureSearch>
```

```
     <search:loginRequirement>
     <search:securityFilterLifespan>
     <search:securityFilterRefreshWaitTimeout>
     <search:authenticationTimeout>
     <search:authorizationTimeout>
     <search:minNumThreads>
     <search:maxNumThreads>
     <search:preserveStaleSecurityFilterOnError>
```

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:lastCrawlDatesMergeRange>

Threshold for merging last crawl dates from different data sources. The default value is 86,400,000.

<search:searchTimeout>

Number of milliseconds allowed for processing each parallel query phase. The default value is 1,200,000.

<search:enableWildcardQueries>

Controls whether question marks (?) and asterisks (*) in queries are used as wildcards or literal characters for matching documents.

Attribute	Value
`enabled`	Set to `true` to enable wildcards, or set to `false` otherwise. Default is `false`. Required.

<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 `true` to enable spelling correction, or set to `false` 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 `true` to enable collection or set to `false` 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 `true` to enable URL submission, or set to `false` 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:securityFilterRefreshWaitTimeout>
<search:authenticationTimeout>
<search:authorizationTimeout>
<search:minNumThreads>
<search:maxNumThreads>
<search:preserveStaleSecurityFilterOnError>
```

<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:securityFilterRefreshWaitTimeout>
Number of milliseconds to block a query for a security filter refresh before returning no results or using an expired security filter, depending on the value of **<search:preserveStaleSecurityFilterOnError>**. The default value is 1000 ms.

<search:authenticationTimeout>
Number of milliseconds for authentication.

<search:authorizationTimeout>
Number of milliseconds for authorization.

<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:preserveStaleSecurityFilterOnError>
Controls the response to queries when an expired security filter is being refreshed. Set to `true` to use the expired security filter, or set to `false` to return no results. The default value is `false`.

When the security filter is expired, Oracle SES triggers a security filter refresh. During the refresh, if there is any error from any data source, then the user's existing security filter is preserved or overwritten, depending on this setting.

Example

This XML document describes the query parameters:

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

```
<search:config xmlns:search="http://xmlns.oracle.com/search"
productVersion="11.2.2.2.0">
   <search:queryConfig>
      <search:maxNumResults>200</search:maxNumResults>
      <search:lastCrawlDatesMergeRange>8640000</search:lastCrawlDatesMergeRange>
      <search:searchTimeout>120000</search:searchTimeout>
      <search:enableWildcardQueries enabled="false"/>
      <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>7</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:securityFilterRefreshWaitTimeout>
            1000</search:securityFilterRefreshWaitTimeout>
         <search:preserveStaleSecurityFilterOnError>
            false</search:preserveStaleSecurityFilterOnError>
      </search:secureSearch>
   </search:queryConfig>
</search:config>
```

queryUIConfig

Query UI configuration enables you to customize the search results.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

Global Settings - Query UI Configuration

XML Description

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

```
<search:queryUIConfig>
  <search:convertTimezone>
     <search:convertTimeZoneAttrs>
        <search:convertTimeZoneAttr>
  <search:defaultTopNDocuments>
  <search:maxTopNDocuments>
  <search:similarDocumentHandling>
  <search:resultsInNewWindow>
  <search:numVisibleGroupTabs>
  <search:groupTabOrder>
     <search:groupTabName>
  <search:displayQueryExpansionMessage>
  <search:fetchBrowseSourceGroupsOnPageload>
  <search:displayQuerySplashPage>
  <search:defaultSourceGroupName>
  <search:suppressedSourceGroupList>
     <search:suppressedSourceGroupName>
  <search:displayErrorPageOnInvalidSourceGroup>
  <search:showSidebarForFreshUser>
  <search:autoLoadTopNResults>
  <search:cacheLogoImage>
     <search:cacheLogoImagePath>
  <search:displayAutoSuggestions>
  <search:minCharBeforeAutoSuggestion>
  <search:absoluteSorting>
     <search:sortableAttrs>
        <search:sortableAttr>
```

Element Descriptions

<search:queryUIConfig>

Describes query UI configuration parameters. It contains these elements:

```
<search:convertTimezone>
<search:defaultTopNDocuments>
<search:maxTopNDocuments>
<search:similarDocumentHandling>
<search:resultsInNewWindow>
<search:numVisibleGroupTabs>
<search:groupTabOrder>
<search:displayQueryExpansionMessage>
<search:fetchBrowseSourceGroupsOnPageload>
<search:displayQuerySplashPage>
<search:defaultSourceGroupName>
<search:suppressedSourceGroupList>
<search:displayErrorPageOnInvalidSourceGroup>
<search:showSidebarForFreshUser>
<search:autoLoadTopNResults>
<search:cacheLogoImage>
<search:displayAutoSuggestions>
<search:minCharBeforeAutoSuggestion>
<search:absoluteSorting>
```

<search:convertTimezone>

Controls whether the date values in query application should be converted to the user's time zone.

Attribute	Value
`enabled`	Set to `true` to convert the date values in query application to the user's time zone, or set to `false` otherwise. Required.

Contains the element `<search:convertTimeZoneAttrs>`.

<search:convertTimeZoneAttrs>

Identifies date attributes that appear in the search results, that need to be converted to user's time zone. It contains one or more `<search:convertTimeZoneAttr>` elements.

<search:convertTimeZoneAttr>

Identifies a date attribute to be converted to user's time zone.

<search:defaultTopNDocuments>

Controls the number of documents to retrieve by default as part of the top N search results.

<search:maxTopNDocuments>

Controls the maximum number of documents to retrieve by default as part of the top N search results.

<search:similarDocumentHandling>

Controls how similar documents should be handled. The available options are:

- `detect`: Detect and display similar documents under the **Similar Documents** link in a search result page.
- `remove`: Detect and remove similar documents from a search result page.
- `disabled`: Do not detect similar documents, that is, similar documents will be displayed in a search result page along with all the other documents.

<search:resultsInNewWindow>

Controls whether clicking the search result link should display search results in a new window, or in the same window.

Attribute	Value
enabled	Set to true to display search results in a new window, or set to false to display search results in the same window. Required.

Note: This setting has no effect if XSLT is used for rendering search results.

<search:numVisibleGroupTabs>

Controls the number of source group tabs to display above the query box. If the number of source groups are more than this value, then you can click the **more >>** link to view the remaining source group tabs.

<search:groupTabOrder>

Controls the ordering of source group tabs that are displayed above the query box. It contains one or more <search:groupTabName> elements.

<search:groupTabName>

Source group tab name to display above the query box.

<search:displayQueryExpansionMessage>

Controls whether to display alternate keywords message in the query application.

Attribute	Value
enabled	Set to true to display alternate keywords message, or set to false otherwise. Required.

<search:fetchBrowseSourceGroupsOnPageload>

Controls whether to fetch the source group list each time the Browse popup window is displayed, otherwise the source group list is fetched only once when the Browse popup window is displayed for the first time, and for the subsequent display of the Browse popup window the cached source group list is displayed.

Attribute	Value
enabled	Set to true to fetch the source group list each time the Browse popup window is displayed, or set to false otherwise. Required.

<search:displayQuerySplashPage>

Controls whether to display the splash page in the query application when the query application is started.

Attribute	Value
enabled	Set to true to display the splash page in the query application when the query application is started, or set to false otherwise. Required.

<search:defaultSourceGroupName>

Default source group to display on the query page. Select **All (System)** to display **All** source group (that is, source group containing all the sources) by default on the query page.

<search:suppressedSourceGroupList>

List of source groups that should not be displayed in the query application.

Contains the element `<search:suppressedSourceGroupName>`.

<search:suppressedSourceGroupName>

Source group name that should not be displayed in the query application.

<search:displayErrorPageOnInvalidSourceGroup>

Controls whether to display an error page if an invalid source group name is specified in the query URL parameters.

Attribute	Value
`enabled`	Set to `true` to display an error page, if an invalid source group is specified in the query URL parameters. If set to `false`, the invalid source group is ignored, and the query uses the default source group, if no other valid source groups are specified. Required.

<search:showSidebarForFreshUser>

Controls whether to display sidebar on the query results page, when the query search is used for the first time (that is, when the cookie is not available).

Attribute	Value
`enabled`	Set to `true` to display sidebar on the query results page, when the query search is used for the first time (that is, when the cookie is not available), or set to `false` otherwise. Required.

<search:autoLoadTopNResults>

Controls whether to automatically load the top-N results on the search results page.

Attribute	Value
`enabled`	Set to `true` to automatically load the top-N results on the search results page. If set to `false`, a button is displayed on the search page, which you can click to load the top-N results. Default is `false`. Required.

<search:cacheLogoImage>

Controls whether to display the cached page logo image.

Attribute	Value
`enabled`	Set to `true` to display the cached page logo image, or set to `false` otherwise. Required.

Contains the element `<search:cacheLogoImagePath>`.

<search:cacheLogoImagePath>

Path of the cached page logo image.

<search:displayAutoSuggestions>

Controls whether to display auto suggestions.

Attribute	Value
enabled	Set to true to display auto suggestions in the query application, or set to false otherwise. Required.

<search:minCharBeforeAutoSuggestion>

The minimum number of characters a user has to enter into the search box so as to display auto suggestion keywords.

See Also: **autoSuggestion** object.

<search:absoluteSorting>

Defines sortable search attributes to show in the **Sort by** list in the query application. Contains the element <search:sortableAttrs>.

<search:sortableAttrs>

Contains one or more <search:sortableAttr> elements.

<search:sortableAttr>

Sortable search attribute to show in the **Sort by** list in the query application.

> **Note:** When the sortable search attributes are provided, the first attribute from sortConditions element specified in relevanceRanking object (at the global level) or defaultSortConditions element specified in queryUISourceGroups object (at the source group level), will be added, along with Relevance, to the **Sort by** list in the query application.

Example

This XML document defines the query UI configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
<search:queryUIConfig>
  <search:convertTimezone enabled="false"/>
  <search:defaultTopNDocuments>100</search:defaultTopNDocuments>
  <search:maxTopNDocuments>300</search:maxTopNDocuments>
  <search:similarDocHandling>detect</search:similarDocHandling>
  <search:resultsInNewWindow>true</search:resultsInNewWindow>
  <search:numVisibleGroupTabs>5</search:numVisibleGroupTabs>
  <search:groupTabOrder/>
  <search:displayQueryExpansionMessage>true</search:displayQueryExpansionMessage>
  <search:useMultipleAlternateKeywords>false</search:useMultipleAlternateKeywords>

<search:fetchBrowseSourceGroupsOnPageload>false</search:fetchBrowseSourceGroupsOnP
ageload>
  <search:displayQuerySplashPage>false</search:displayQuerySplashPage>
  <search:absoluteSorting>
     <search:sortableAttrs>
        <search:sortableAttr>price</search:sortableAttr>
        <search:sortableAttr>author</search:sortableAttr>
     </search:sortableAttrs>
  </search:absoluteSorting>
</search:queryUIConfig>
</search:config>
```

queryUIFacets

Facets are used to refine the search results. The `queryUIFacets` object contains the configuration settings for controlling the display properties of facets in the query application.

Object Type

Universal

State Properties

Property	Value
`status`	`ACTIVE` `INACTIVE`

Supported Operations

```
activate
deactivate
getState
update
export
```

Administration GUI Page

Global Settings - Configure Facets

XML Description

The `<search:queryUIFacets>` element describes facets display configurations:

```
<search:queryUIFacets>
   <search:numVisibleFacets>
   <search:facetTrees>
      <search:facetTree>
```

Element Descriptions

<search:queryUIFacets>
Contains one or more `<search:facetTrees>` elements.

Attribute	Value
`enabled`	Set to `true` to enable facets, or set to `false` otherwise. Required.

<search:numVisibleFacets>
Number of facets to display on the query application screen.

<search:facetTrees>
Contains one or more `<search:facetTree>` elements.

<search:facetTree>
Controls the following display properties of a facet:

Attribute	Value
name	Name of the facet.
enabled	Set to true to enable, that is, to display this facet, or set to false otherwise.
sortBy	Controls the sorting order of documents matching this facet. The following are the available sorting criteria: COUNT_DESCENDING COUNT_ASCENDING ALPHA_DESCENDING (for String type facet only) ALPHA_ASCENDING (for String type facet only) TREE_STRUCTURE (for Number or Date type facet only)
minDocPerNode	The minimum number of documents that should contain the match for a facet node of this facet. If the documents matching a facet nodes are less than minDocPerNode value, then that facet node is not displayed in the query application.
numOfVisibleValues	Number of facet nodes to display for the facet in the query application. If the number of facet nodes returned for a search result are more than this value, then you need to click the **More ...** link to see the next set of facet nodes in the facet panel. The number of additional facet nodes to display by clicking the **More ...** link is controlled by the value provided in this attribute. For example, if numOfVisibleValues is set to 5, then each time **More ...** link is clicked, the additional five facet nodes are displayed.

Example

This XML document configures display propertied of facets:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:queryUIFacets enabled="true">
      <search:numVisible>4</search:numVisible>
      <search:facetTrees>
            <search:facetTree name="Author" enabled="true"
                              sortBy="COUNT_DESCENDING" numOfVisibleValues="5"
                              minDocPerNode="1"/>
            <search:facetTree name="LastModified" enabled="false"
                              sortBy="TREE_STRUCTURE" numOfVisibleValues="10"
                              minDocPerNode="1"/>
      </search:facetTrees>
   </search:facetUIConfig>
</search:config>
```

queryUISourceGroups

This object contains the source group related configuration settings, such as, facets, cluster trees, top-N sortable attributes, top-N groupable attributes, and sortable attributes for absolute sort.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

Global Settings - Configure Source Groups

XML Description

The `<search:queryUISourceGroups>` element describes source group specific configuration settings, such as, facets, cluster trees, top-N sortable attributes, top-N groupable attributes, and sortable attributes for absolute sorting:

```
<search:queryUISourceGroups>
   <search:sourceGroups>
      <search:sourceGroup>
         <search:facetTrees>
            <search:facetTree>
            <search:all>
            <search:none>
         <search:clusterTrees>
            <search:clusterTree>
            <search:all>
            <search:none>
         <search:topNSortableAttrs>
            <search:topNSortableAttr>
            <search:defaults>
            <search:none>
         <search:topNGroupableAttrs>
            <search:topNGroupableAttr>
            <search:defaults>
            <search:none>
         <search:absoluteSorting>
            <search:defaultSortConditions>
               <search:sortCondition>
            <search:sortableAttrs>
               <search:sortableAttr>
```

Element Descriptions

<search:queryUISourceGroups>
Contains one or more `<search:sourceGroups>` elements.

<search:sourceGroups>
Collection of source groups. Contains one or more `<search:sourceGroup>` elements.

<search:sourceGroup>
A source group specific configurations. Contains one or more `<search:facetTrees>` elements.

<search:facetTrees>
Facets related to the source group. Contains `<search:all>` or `<search:none>` or one or more `<search:facetTree>` elements.

<search:facetTree>
Facet to display for the source group.

<search:all>
Indicates that all the facets must be displayed for the source group in the query application.

<search:none>
Indicates that no facets should be displayed for the source group in the query application.

<search:clusterTrees>
Cluster trees to display in the query application for the source group. Contains `<search:all>` or `<search:none>` or one or more `<search:clusterTree>` elements.

<search:clusterTree>
Name of the cluster tree to display for the source group in the query application.

<search:all>
Indicates that all the cluster trees must be displayed for the source group in the query application.

<search:none>
Indicates that no cluster trees should be displayed for the source group in the query application.

<search:topNSortableAttrs>
Contains one or more `<search:topNSortableAttr>` elements.

<search:topNSortableAttr>
Attribute to show in the Sort by list in the query application for the source group.

<search:defaults>
Indicates that all the default attributes should be displayed in the Sort by list in the query application for the source group. The default top-n sortable attributes are:

- Author
- File Format
- Title
- Relevance

- Path
- Language
- Date

<search:none>
Indicates that no attributes should be displayed in the Sort by list in the query application for the source group.

> **Note:** The system attribute **Relevance** will still be shown in the **Sort by** list.

<search:topNGroupableAttrs>
Contains one or more `<search:topNGroupableAttr>` elements.

<search:topNGroupableAttr>
Attribute to show in the Group by list in the query application for the source group.

<search:defaults>
Indicates that all the default attributes should be displayed in the Group by list in the query application for the source group. The default top-n groupable attributes are:

- (none)
- Author
- File Format
- Source
- Date

<search:none>
Indicates that no attributes should be displayed in the Group by list in the query application for the source group.

> **Note:** The system attributes **(none)** and **Source** will still be shown in the Group by list.

<search:absoluteSorting>
Contains `<search:defaultSortConditions>` and `<search:sortableAttrs>` elements.

<search:defaultSortConditions>
Defines the default sort conditions for the query application. Contains one or more `<search:sortCondition>` elements.

<search:sortCondition>
Defines a default sort condition.

Attribute	Value
`name`	Name of the sortable attribute.
`order`	Sort order of the specified attribute. The available options are `ascending` and `descending`. Default is `ascending`.

Attribute	Value
type	Sortable attribute type. For sortable search attribute, specify ATTRIBUTE, and for sortable system attribute, specify SYSTEM. Default is ATTRIBUTE.

<search:sortableAttrs>

Defines the sortable search attributes to display in the Sort by list in the query application. Contains one or more <search:sortableAttr> elements.

<search:sortableAttr>

Sortable attribute to display in the Sort by list in the query application.

Attribute	Value
name	Name of the sortable attribute.

> **Note:** When the sortable search attributes are provided, the first attribute from defaultSortConditions element specified in queryUISourceGorups object (at the source group level) or sortConditions element specified in relevanceRanking object (at the global level), will be added, along with Relevance, to the **Sort by** list in the query application.

Example

This XML document defines the source group sourceGroup1:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   </search:queryUISourceGroups>
      <search:sourceGroup name="sourceGroup1">
         <search:facetTrees>
            <search:facetTree name="Tree 1"/>
            <search:facetTree name="Tree 2"/>
            <search:facetTree name="Tree 3"/>
         </search:facetTrees>
         <search:clusterTrees>
            <search:all/>
         </search:clusterTrees>
         <search:topNSortableAttrs>
            <search:topNSortableAttr name="sort attribute 1"/>
            <search:topNSortableAttr name="sort attribute 2"/>
         </search:topNSortableAttrs>
         <search:topNGroupableAttrs>
            <search:topNGroupableAttr name="group attribute 1"/>
         </search:topNGroupableAttrs>
         <search:absoluteSorting>
            <search:defaultSortConditions>
               <search:sortCondition name="price" order="ascending"/>
               <search:sortCondition name="LastModifiedDate" order="descending"/>
            </search:defaultSortConditions>
            <search:sortableAttrs>
               <search:sortableAttr name="price"/>
               <search:sortableAttr name="author"/>
            </search:sortableAttrs>
```

```
                </search:absoluteSorting>
            </search:sourceGroup>
        </search:queryUISourceGroups>
    </search:config>
```

relevanceRanking

Relevance ranking controls the importance given to various document attributes when ordering the search results. By customizing the ranking rules, you can produce more relevant search results for your enterprise.

In Oracle SES 11.1.2.2 and earlier releases, these parameters were stored in a file named `ranking.xml`.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

Search - Forced Ranking

Search - Inline Result Grouping

Search - Ranking Factor

XML Description

The `<search:relevanceRanking>` element describes the relevance ranking of search attributes:

```
<search:relevanceRanking>
   <search:defaultFactors>
      <search:defaultFactor>
         <search:name>
         <search:weight>
   <search:customFactors>
      <search:customFactor>
         <search:attributeName>
         <search:attributeType>
         <search:factorType>
            <search:queryFactor>
               <search:weight>
            <search:staticFactor>
               <search:matches>
                  <search:match>
                     <search:value>
                     <search:weight>
   <search:docScoreFactors>
      <search:docScoreFactor>
         <search:attributeName>
         <search:weight>
   <search:sortConditions>
      <search:sortCondition>
   <search:queryModels>
```

```
      <search:macros>
         <search:macro>
      <search:queryModel>
         <search:triggers>
            <search:trigger>
         <search:includeQueries>
            <search:includeQuery>
         <search:includeUrls>
            <search:includeUrl>
   <search:resultGroupings>
      <search:maxResultGroupsReturned>
      <search:positions>
         <search:position>
      <search:resultGroupingList>
         <search:resultGrouping>
            <search:name>
            <search:key>
            <search:value>
            <search:teasers>
```

Element Descriptions

<search:relevanceRanking>

Contains these elements:

```
<search:defaultFactors>
<search:customFactors>
<search:docScoreFactors>
<search:sortConditions>
<search:queryModels>
<search:resultGroupings>
```

<search:defaultFactors>

Sets the weights for the default attributes used for ranking. It contains one or more `<search:defaultFactor>` elements.

Attribute	Value
`enabled`	Set to `true` to enable default factors (default), or set to `false` otherwise. Required.

<search:defaultFactor>

Identifies a default search attribute and its weight. It contains these elements:

```
<search:name>
<search:weight>
```

The following table lists the default attributes and weights:

Attribute	Weight
`Title`	`High`
`Description`	`Medium`
`Reftext`	`High`
`Keywords`	`Medium`
`Subject`	`Low`

Attribute	Weight
`Author`	`Medium`
`H1headline`	`Low`
`H2headline`	`Very low`
`Url`	`Low`
`Urldepth`	`High`
`Language Match`	`High`
`Linkscore`	`High`

<search:name>
Name of the attribute, such as `Title` or `Description`.

<search:weight>
Contains the weight assigned to an attribute: `very high`, `high`, `medium`, `low`, `very low`, and `none`. If the weight is not specified, the default weight for the attribute is used.

<search:customFactors>
Adds other attributes for ranking. It contains one or more `<search:customFactor>` elements.

<search:customFactor>
Describes an attribute used for ranking. Any indexed search attribute can be a custom ranking attribute. This element contains these child elements:

```
<search:attributeName>
<search:attributeType>
<search:factorType>
```

<search:attributeName>
The exact name of a search attribute defined in Oracle SES. This name is case-insensitive.

<search:attributeType>
The data type of the attribute. Only `String` is supported.

<search:factorType>
Identifies the type of ranking. It contains one of these elements:

```
<search:queryFactor>
<search:staticFactor>
```

<search:queryFactor>
Matches the attribute value against query terms. For example, if a custom attribute has the value "Terry Francona," then a query for "Terry Francona" is given the relevancy ranking of the attribute.

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

<search:weight>
Contains the weight assigned to an attribute: `very high`, `high`, `medium`, `low`, `very low`, and `none`. If the weight is not specified, the default weight for the attribute is used.

<search:staticFactor>
Matches the attribute value against an attribute of the documents. For example, assume a company identifies its documents as good or poor and defines a custom

search attribute for quality. If a custom attribute for quality ranks good documents `very high` and poor documents `low`, then a good document appears higher than a poor document in the list of search results.

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

<search:matches>
Contains one or more `<search:match>` elements.

<search:match>
Identifies a matching search attribute and value. It contains these elements:

```
<search:value>
<search:weight>
```

<search:value>
The value of the search attribute specified in `<search:name>` being given a weight.

<search:weight>
Contains the weight assigned to an attribute: `very high`, `high`, `medium`, `low`, `very low`, and `none`. If the weight is not specified, the default weight for the attribute is used.

<search:docScoreFactors>
Sets the weights for the document score attributes used for ranking. It contains one or more `<search:docScoreFactor>` elements.

<search:docScoreFactor>
Describes a document score factor. It contains these elements:

```
<search:attributeName>
<search:weight>
```

<search:attributeName>
Name of the document score attribute.

<search:weight>
Weight of the document score attribute. This is a float value.

<search:sortConditions>
Defines the default sort criteria for the query application. Contains one or more `<search:sortCondition>` elements.

<search:sortCondition>
This can be either a sortable search attribute name or a system defined sortable attribute name, such as, `RELEVANCE` and `ABSOLUTE_DATE`. It can have a sort order of either ascending or descending. This sort condition is used as a default sort criteria for ordering the search results.

Attribute	Value
`order`	Sort order of the specified attribute. The available options are `ascending` and `descending`. Default is `ascending`.
`type`	Sortable attribute type. For sortable search attribute, specify `ATTRIBUTE`, and for sortable system attribute, specify `SYSTEM`. Default is `ATTRIBUTE`.

<search:queryModels>
Contains these elements:

```
<search:macros>
<search:queryModel>
```

<search:macros>
Contains one or more `<search:macro>` elements.

<search:macro>
Defines a macro. It contains the following attributes.

Attribute	Value
define	Definition of a macro. A macro has the syntax of `macro_name(parameter1, parameter2, ..., parameterN)`. Macro parameters must be specified using a single letter. Parameters can be referenced in the macro definition by prefixing them with a % sign. Macros can be used in both triggers and query expressions.

The following macros are provided by Oracle SES by default.

Macro	Purpose
`<macro define='prefix(P)'> ^%P </macro>`	Prefix match (used for triggers)
`<macro define='suffix(S)'> %S$ </macro>`	Suffix match (used for triggers)
`<macro define='exact(E)'> ^%E$ </macro>`	Exact match (used for triggers)

<search:queryModel>
Defines a query model. It contains these elements:

```
<search:triggers>
<search:includeQueries>
<search:includeUrls>
```

It contains the following attributes.

Attribute	Value
name	Name of the query model.
inherit	Name of the inherited query model (optional). Query models support single inheritance. The most common usage of this is to support default models for common query-independent filter expressions.

Example of query model inheritance:

```
<!-- Parent query model -->
<queryModel name="default">
   <includeQueries>
      <includeQuery>mimetype:=text/html or mimetype:=application/pdf or
mimetype:=text/plain</includeQuery>
   </includeQueries>
</queryModel>

<!-- Child query model -->
<queryModel name='whitepaper' inherit='default'>
   <triggers>
      <trigger>((\w+\s+)+)whitepapers?</trigger>
      <trigger>((\w+\s+)+)white\s+papers?</trigger>
   </triggers>
   <includeQueries>
```

```
      <includeQuery>TAG:$1 and TAG:"white papers"</includeQuery>
   </includeQueries>
</queryModel>
```

Here, "whitepaper" query model inherits "default" query model. The parent model (default) is connected with the child model (whitepaper) using AND operator. This has the effect of limiting the whitepaper query model hits to mime-types of html, pdf, and plain text. A parent query model cannot contain any triggers; it must contain only query-independent filter expressions.

<search:triggers>

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

<search:trigger>

Defines a trigger to be issued on a user query string. It should contain a regular expression confirming to POSIX standard. For example, use `<trigger>oracle.*</trigger>` for matching all the query terms starting with the word "oracle".

<search:includeQueries>

Contains one or more `<search:includeQuery>` elements that define a set of query expressions to use to generate and score the top-ranking hits.

<search:includeQuery>

Defines a query expression to generate and score the top-ranking hits. For example:

```
<queryModel name='titles'>
   <triggers>
      <trigger>Title:(.*)</trigger>
   </triggers>
   <includeQueries>
      <includeQuery>Title:"$1" and Host:"oracle.com"</includeQuery>
   </includeQueries>
</queryModel>
```

Here, whenever an attribute search is done for a `Title`, the top results will be shown only from the documents present on the host `oracle.com`.

When one or more query expressions are specified, the order of defining the query expressions correspond to their weights, that is, a query expression defined first has a higher rank as compared to query expressions defined subsequently. The query expressions are evaluted together using OR operation. A query expression should be an attribute only query. For example,

```
<queryModel name='gym'>
   <triggers>
      <trigger>gym</trigger>
   </triggers>
   <includeQueries>
      <includeQuery>TAG:"gym" AND TAG:"landing page"</includeQuery>
      <includeQuery>TAG:"gym" AND TAG:"news"</includeQuery>
   </includeQueries>
</queryModel>
```

Here, the query model assumes that someone had tagged the gym landing page as well as news about the gym. It ranks the gym landing page first followed by news pages by creating the following Oracle Text query:

```
(( (MDATA(TAG,gym)) AND (MDATA(TAG,landing page)) )*1.0)|
(( (MDATA(TAG,gym)) AND (MDATA(TAG,news)) )*0.99)
```

<search:includeUrls>
Contains one or more `<search:includeUrl>` elements that define a set of URLs to be shown on top of search results in the order of their definition.

<search:includeUrl>
URL to be shown on top of search results in the order of its definition.

<search:resultGroupings>
Describes result grouping configuration. It contains these elements:

```
<search:maxResultGroupsReturned>
<search:positions>
<search:resultGroupingList>
```

<search:maxResultGroupsReturned>
Maximum number of result groupings to show in search results.

<search:positions>
Contains one or more `<search:position>` elements that define the position of result groupings to display in search results.

<search:position>
Postion of a result grouping to show in search results.

<search:resultGroupingList>
Contains one or more `<search:resultGrouping>` elements that define a list of result groupings configuration.

<search:resultGrouping>
Defines a result grouping configuration. It contains these elements:

```
<search:name>
<search:key>
<search:value>
<search:teasers>
```

<search:name>
Name of result grouping.

<search:key>
Type of result grouping. It can be either "Source Group" or "URL".

<search:value>
Value of the result grouping name, that is, either the source group name or the URL. This is optional.

<search:teasers>
Number of teaser links to show within the result grouping. If the number of teasers is set to 0, then the result grouping will not be shown in search results.

Example

This XML document describes relevance ranking configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:relevanceRanking>

      <!-- defaultFactors -->
      <search:defaultFactors enabled="true">
         <search:defaultFactor>
            <search:name>TITLE</search:name>
            <search:weight>MEDIUM</search:weight>
         </search:defaultFactor>
      </search:defaultFactors>

      <!-- sortConditions -->
      <search:sortConditions>
         <search:sortCondition order="ascending" type="attribute">Attribute_
1</search:sortCondition>
         <search:sortCondition order="descending"
type="system">RELEVANCE</search:sortCondition>
      </search:sortConditions>

      <!-- resultGroupings -->
      <search:resultGroupings>
         <search:maxResultGroupsReturned>10</search:maxResultGroupsReturned>
         <search:positions>
            <search:position>2</search:position>
            <search:position>3</search:position>
            <search:position>7</search:position>
         </search:positions>
         <search:resultGroupingList>
            <search:resultGrouping>
               <search:name>oracle</search:name>
               <search:key>url</search:key>
               <search:value>http://www.oracle.com</search:value>
               <search:teasers>2</search:teasers>
            </search:resultGrouping>
         </search:resultGroupingList>
      </search:resultGroupings>

      <!-- queryModels -->
      <search:queryModels>
         <search:queryModel name="default">
            <search:includeQueries>
               <search:includeQuery>mimetype:=text/html or
mimetype:=application/pdf or mimetype:=text/plain</search:includeQuery>
            </search:includeQueries>
         </search:queryModel>
         <search:queryModel name="ses" inherit="default">
            <search:triggers>
               <search:trigger>@EXACT(secure enterprise search)</search:trigger>
               <search:trigger>@EXACT(ses)</search:trigger>
               <search:trigger>@EXACT(secure search)</search:trigger>
            </search:triggers>
            <search:includeQueries>
               <!-- show landing pages first by tag -->
               <search:includeQuery>TAG:"secure enterprise search" AND
TAG:"landing page"</search:includeQuery>
               <!-- show downloads by tag -->
               <search:includeQuery>TAG:"secure enterprise search" AND
TAG:"downloads"</search:includeQuery>
            </search:includeQueries>
         </search:queryModel>
      </search:queryModels>

   </search:relevanceRanking>
</search:config>
```

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="11.2.2.2.0"
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="11.2.2.2.0"
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:properties>
         <search:property>
      <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:properties>
<search:translations>
<search:lovEntries>
```

<search:name>
Name of the search attribute.

<search:type>
Data type of the attribute values. Set to STRING, NUMBER, or DATE.

<search:properties>
Defines a list of attribute properties. Contains one or more <search:property> elements.

<search:property>
Defines an attribute property. Currently, the only property supported is sortable, which takes the value of either true or false.

Attribute	Value
name	You can only specify sortable.
value	Specify true to enable sorting, else specify false.

<search:translations>
Provides a display name. See "Providing Translations of Object Names" on page 2-7.

<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.

<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="11.2.2.2.0"
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>
```

singleSignOnSetting

The Single Sign-On (SSO) settings are used to configure SSO types, such as, Oracle Access Manager (OAM), Oracle Single Sign-On (OSSO), and Windows Native Authentication (WNA).

OAM is supported by both the admin application and the query application, while OSSA and WNA are supported only by the query application.

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
deactivate
export
exportAll
exportList
getAllObjectKeys
getAllStates
getState
getStateList
update
updateAll
```

Administration GUI Pages

Global Settings - Configure Single Sign-On

XML Description

The `<search:singleSignOnSettings>` element describes the configuration settings for SSO:

```
<search:singleSignOnSettings>
   <search:singleSignOnSetting>
      <search:name>
      <search:parameters>
      <search:parameters>
         <search:parameter>
         <search:value>
```

Element Descriptions

<search:singleSignOnSettings>
Contains one or more `<search:singleSignOnSetting>` elements, which contain the SSO options that can be activated.

<search:singleSignOnSetting>
Contains SSO settings for an SSO type. It contains these elements:

```
<search:name>
<search:description>
<search:parameters>
```

<search:name>
The SSO type. The supported values are `OAM`, `OSSO`, and `WNA`.

<search:description>
Description of the SSO type.

<search:parameters>
Contains one or more `<search:parameter>` elements.

<search:parameter>
Name of the SSO configuration parameter. These parameters vary for different SSO types. The supported parameters are:

- **For OAM**:

  Query invalid session return URL – URL to display when the query application session expires.

  Query logout return URL – URL to display after logging out of the query application.

  Admin logout return URL – URL to display after logging out of the administration application.

- **For OSSO**:

  Hint cookie enabled – whether hint cookie should be enabled (true/false).

  Hint cookie name – name of the hint cookie.

  Query invalid session return URL – URL to display when the query application session expires.

  Query logout return URL – URL to display after logging out of the query application.

- **For WNA**:

  There are no parameters for WNA SSO type.

<search:value>
Value of the SSO configuration parameter.

Example

The following is the default XML for Single Sign-On configuration settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
  <singleSignOnSettings>
     <singleSignOnSetting>
        <search:name>OAM</search:name>
     </singleSignOnSetting>
     <singleSignOnSetting>
        <search:name>OSSO</search:name>
     </singleSignOnSetting>
     <singleSignOnSetting>
        <search:name>WNA</search:name>
     </singleSignOnSetting>
  </singleSignOnSettings>
</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 2-11

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
deactivate
delete
deleteAll
deleteList
export
exportAll
exportList
getAllObjectKeys
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="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:skinBundles>
      <search:skinBundle>
         <search:name>acme</search:name>
         <search:isDefault>false</search:isDefault>
         <search:linkedVersion>11.2.2.2.0</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 `true` if the password is encrypted, or set to `false` 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: STRING, NUMBER, or DATE.
isAvailable	Identifies whether the attribute is currently available in the federated source: true if it is available, or false otherwise.
isMandatory	Identifies whether retrieval of the attribute is mandatory: true if it must be listed in the <search:retrievedAttrs> element, or false 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: STRING, NUMBER, or DATE. (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: STRING, NUMBER, or DATE. (Required)
isAvailable	Identifies whether the remote attribute is currently available in the federated source: true if it is available, or false otherwise.

Example 2–1   Federated Source Description

This XML document describes a federated source:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
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: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:followSymlinks>
   <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 `true` to use the display URL, or set to `false` 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 2-151.

<search:authorizationPlugin>

Describes the authorization plug-in. See "XML Description: User-Defined Sources" on page 2-146.

<search:boundaryRules>

Describes the boundary rules for the source. See "XML Description: Web Sources" on page 2-151.

<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:charSetDetection>
<search:defaultCharSet>
<search:servicePipeline>
<search:indexNullTitleFallback>
```

```
<search:badTitles>
<search:logLevel>
<search:followSymlinks>
```

See the **<search:crawlerSettings>** for Web sources on page 2-155 for description for all these elements, except the `<search:followSymlinks>` element, which is described as follows.

<search:followSymlinks>
Contains `true` to prevent the crawler from following links to the absolute path, or `false` otherwise. The default value is `true`.

Applies only to file sources on Linux and UNIX systems.

<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 Supported by Oracle SES" 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="11.2.2.2.0"
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:attributeMapping>
               <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
Federated User Authorization Cache
Lotus Notes
Microsoft Exchange)
Microsoft SharePoint 2007
```

```
NTFS
Oracle Calendar
Oracle Collaboration Suite E-Mail
Oracle Content Database
Oracle Content Database (JDBC)
Oracle Content Server
Oracle E-Business Suite
Oracle Fusion
Oracle WebCenter
Push Feed
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 2-151.

<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 *ses_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 `<search:value>` is encrypted. Set to `true` if the value is encrypted, or set to `false` 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 2-151.

<search:attributeMappings>

Maps the document attributes to search attributes. See "XML Description: File Sources" on page 2-141.

<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 2-151.

<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 Supported by Oracle SES".

Example 2–3 User-Defined Source Description for Oracle Content Database source

The following XML document describes an Oracle Content Database source.

```
<?xml version="1.0"?>
<search:config productVersion="11.2.2.2.0"
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>
```

Example 2–4 User-Defined Source Description for Push Feed source

The following XML document describes a Push Feed source:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:sources>
      <search:userDefinedSource>
         <search:name>pshsrc3</search:name>
         <search:sourceTypeName>Push Feed</search:sourceTypeName>
         <search:aclPolicy>
            <search:documentLevelACL/>
         </search:aclPolicy>
         <search:authorizationPlugin>

<search:managerClassName>oracle.search.plugin.security.auth.db.DBAuthManager</sear
ch:managerClassName>

<search:jarFilePath>oracleapplications/DBCrawler.jar</search:jarFilePath>
            <search:parameters>
               <search:parameter name="Authorization Database Connection String">
                  <search:value>DBCONNECTSTR1</search:value>
                  <search:description>JDBC connection string for the
database</search:description>
               </search:parameter>
               <search:parameter name="Authorization Query">
```

```
                  <search:value>select EQ_GROUPS1 from pushtestuseracl where
username like upper(?)</search:value>
                  <search:description>SQL query to retrieve values of all the
security attributes for a given user. The user ID in the WHERE clause should be
specified as '?'. For example, SELECT attr1, attr2 FROM table1, table2 WHERE
table1.f1=table2.f2 AND table1.user=?.</search:description>
               </search:parameter>
               <search:parameter name="Authorization User ID Format">
                  <search:value>nickname</search:value>
                  <search:description>Format of user ID to be used in the
authorization query. This format should be one of the supported authentication
attributes of the active ID plugin. The canonical form will be used if format is
not specified.</search:description>
               </search:parameter>
               <search:parameter name="JDBC Driver Class">
                  <search:value>oracle.jdbc.driver.OracleDriver</search:value>
                  <search:description>JDBC driver class to connect to the
database. For example, oracle.jdbc.driver.OracleDriver</search:description>
               </search:parameter>
               <search:parameter name="Password">
                  <search:value
encrypted="true">b911a0fa2b08f209c53f50131339e06b62526a22cd205807</search:value>
                  <search:description>Password to connect to the
database</search:description>
               </search:parameter>
               <search:parameter name="Single Record Query">
                  <search:value>false</search:value>
                  <search:description>Enter true if the query returns single
record for each user with attribute values separated by spaces. Else, enter
false.</search:description>
               </search:parameter>
               <search:parameter name="User ID">
                  <search:value>scott</search:value>
                  <search:description>User ID to connect to the
database</search:description>
               </search:parameter>
            </search:parameters>
         </search:authorizationPlugin>
         <search:securityAttrs>
            <search:securityAttr type="GRANT">EQ_GROUPS1</search:securityAttr>
         </search:securityAttrs>
         <search:parameters>
            <search:parameter name="Attachment Auth Type">
               <search:value>NATIVE</search:value>
               <search:description>Authentication Type for
Attachments</search:description>
            </search:parameter>
            <search:parameter name="Attachment Realm">
               <search:description>Realm for attachments</search:description>
            </search:parameter>
            <search:parameter name="HTTP Error Log URL">
               <search:description>The HTTP URL where the status feeds are sent
after batch feed is processed.</search:description>
            </search:parameter>
            <search:parameter name="Scratch Directory">
               <search:description>Scratch Directory</search:description>
            </search:parameter>
            <search:parameter name="Security Attributes">
               <search:value>EQ_GROUPS1,true</search:value>
               <search:description>Security attributes, comma separated list of
```

```
name, (grant/deny)</search:description>
            </search:parameter>
            <search:parameter name="Source Password">
               <search:value
encrypted="true">42fde84db62dcd4eccfe438d941fe0f854e7580f584926e2</search:value>
               <search:description>Password for fetching
Attachments</search:description>
            </search:parameter>
            <search:parameter name="Source Username">
               <search:value>aime</search:value>
               <search:description>Username for fetching
Attachments</search:description>
            </search:parameter>
            <search:parameter name="Trusted Entity Password">
               <search:value
encrypted="true">ef38accc4d3185115bc12913edb3570cf68fcdb90f15eb43</search:value>
               <search:description>Trusted Entity Password</search:description>
            </search:parameter>
            <search:parameter name="Trusted Entity Username">
               <search:value>testuser1</search:value>
               <search:description>Trusted Entity Username</search:description>
            </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:charsetDetection>
            <search:defaultCharSet>
            <search:servicePipeline>
               <search:pipelineName>
            <search:indexNullTitleFallback>
            <search:badTitles>
               <search:badTitle>
            <search:honorRobotsExclusion>
            <search:sitemap>
            <search:indexDynamicPages>
            <search:urlRewriter>
               <search:urlRewriterClass>
               <search:urlRewriterJar>
            <search:httpCharSetOverride>
            <search:cookies>
               <search:cookieContentInLog>
               <search:maxCookieSize>
               <search:maxCookies>
               <search:maxCookiesPerHost>

            <search:agentString>
            <search:duplicateDetection>
            <search:connections>
               <search:timeout>
               <search:retries>
               <search:retryInterval>
            <search:logLevel>
         <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 or 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 2-146.

<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:metataMappings>` 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:charsetDetection>
<search:defaultCharSet>
<search:servicePipeline>
<search:indexNullTitleFallback>
<search:badTitles>
<search:honorRobotsExclusion>
<search:sitemap>
<search:indexDynamicPages>
```

```
<search:urlRewriter>
<search:httpCharSetOverride>
<search:cookies>
<search:logLevel>
```

<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 `true` to enable language detection, or set to `false` 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
`haslimit`	Controls whether the search limit is enforced. Set to `true` to impose the limit, or set to `false` 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
`enabled`	Set to `true` to retain the cache, or set to `false` otherwise. (Required)

<search:charsetDetection>
Contains a value of `true` to enable automatic character set detection, or `false` to disable it. The default value is `true`. This parameter can be set at the global level.

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

<search:pipelineName>

Contains the name of a pipeline.

<search:indexNullTitleFallback>

Controls whether the default title is included in the index for documents with null titles:

- indexForAll: Includes the default title in the index. (Default)
- noIndex: Does not include the default title in the index.

<search:badTitles>

Contains one or more <search:badTitle> elements. This parameter can be set at the global level.

<search:badTitle>

Contains an exact character string for a document title that the crawler omits from the index. These bad titles are defined by default:

```
PowerPoint Presentation
Slide 1
```

<search:honorRobotsExclusion>

Controls visits by robots to the Web site.

Attribute	Value
enabled	Set to true to exclude robots, or set to false otherwise.

<search:sitemap>

Controls the Sitemap processing. The available options are:

- SITEMAP_ONLY: Crawler indexes only those URLs extracted from the Sitemap files. The non-Sitemap URLs that are specified as Starting URLs in the **Basic Settings** page for a Web source are also indexed, but not crawled. All the Sitemap URLs are not crawled further down the URL hierarchy.
- SITEMAP_PREFERRED: If a Sitemap URL is present in robots.txt file or Sitemap.xml file, or at least one Sitemap URL is specified as a Starting URL in the **Basic Settings** page for a Web source, then the crawling is done according to the **Sitemap Only Crawl** option.

  When no Sitemap URL is found, then the regular crawling is done, that is, each Starting URL is crawled further down the URL hierarchy till the last level.
- SITEMAP_FULL: Regular crawling is done for all the Sitemap URLs present in robots.txt file, Sitemap.xml file, and all the Starting URLs, including the Sitemap URLs, that are specified in the **Basic Settings** page for a Web source. Thus, each Sitemap URL as well as non-Sitemap URL is crawled further down the URL hierarchy till the last level.

<search:indexDynamicPages>

Controls whether dynamic pages are crawled and indexed.

Attribute	Value
enabled	Set to true to crawl dynamic pages, or set to false 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 true to use the URL Rewriter, or set to false otherwise.

<search:urlRewriterClass>

Contains the class name of the URL Rewriter.

<search:urlRewriterJar>

Contains the absolute 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 true to exclude robots, or set to false 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 true to enable cookies (default), or false otherwise.

<search:cookieContentInLog>

Controls whether information about cookies appears in the log file.

Attribute	Value
enabled	Set to true to log cookie messages, or set to false 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:agentString>
Contains the browser agent string presented to the Web server. The default value is "Oracle Secure Enterprise Search". Applies only to Web and Portal sources.

<search:duplicateDetection>
Contains a value of `true` to enable duplicate detection during a Web crawl, or `false` to disable it. The default value is `true`.

<search:connections>
Sets limits on a connection to Web and Portal sources. It contains these elements:

```
<search:timeout>
<search:retries>
<search:retryInterval>
```

<search:timeout>
Contains the maximum number of milliseconds to make a connection to a data source. The default value is 10.

<search:retries>
Contains the maximum number of connection attempts to a data source. The default value is 10.

<search:retryInterval>
Contains the number of milliseconds between connection retry attempts. The default value is 5.

<search:logLevel>
Contains the log level for the crawler. The following are the valid log levels:

Logging Level	Description
`TRACE`	Trace messages
`DEBUG`	Debug messages
`INFO`	Informational messages (Default)
`WARN`	Warning messages
`ERROR`	Error messages
`FATAL`	Fatal messages

<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 Supported by Oracle SES".

<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 `<search:password>` is encrypted. Set to `true` if the password is encrypted, or set to `false` 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
encrypted	Indicates whether the value of `<search:value>` is encrypted. Set to `true` if the value is encrypted, or set to `false` 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
enabled	Controls use of OracleAS Single Sign-On for authentication. Set to `true` to enable Single Sign-On, or `false` 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
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:userAgent>

Contains an authentication value that overrides the default User Agent value for OracleAS Single Sign-On. The default value is null.

Example 2–5 Sample Web Source Description

This XML document describes a sample Web source.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:sources>
      <search:webSource>
         <search:name>websource_1</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: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: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>testsso</search:username>
            <search:password encrypted="false">
               password
            </search:password>
         </search:ssoAuthentication>
      </search:webSource>
   </search:sources>
</search:config>
```

Example 2–6 Sample Web Source Description for Configuring Sitemap

This XML document describes Sitemap configuration for a Web source.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:sources>
      <search:webSource>
         <search:name>websource_2</search:name>
         <search:crawlerSettings>
            <search:sitemap>SITEMAP_ONLY</search:sitemap>
         </search:crawlerSettings>
      </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 2-7.

<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="11.2.2.2.0"
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 *ses_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).

<search: description>
Description of the parameter.

Example

This XML document describes the `Oracle Content Database` source type:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
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: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>
```

storageArea

A storage area is equivalent to an Oracle ASSM tablespace that must be created by the Oracle SES database administrator. The `storageArea` object just registers the existing tablespace with Oracle SES.

> **See Also:** "Parallel Querying and Index Partitioning" in *Oracle Secure Enterprise Search Administrator's Guide*

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>
```

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:name>
Name of the storage area. (Required)

Enter the name of an existing ASSM tablespace and specify `PARTITION` for the usage type. An ASSM (Automatic Segment Space Management) tablespace can be created with the SQL `CREATE TABLESPACE` clause `EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO`.

The default tablespaces for Oracle SES are SEARCH_DATA, SEARCH_INDEX, and SEARCH_TEMP.

<search:description>
Description of the storage area. (Required)

<search:usage>
A storage type. (Required)

- `PARTITION`: Stores document index.
- `CACHE_FILE`: Stores secure cache. You cannot create or delete the cache file storage area.
- `CRAWLER`: Stores tokens for index. This storage type is used by the Push crawler.
- `SYSTEM`: Stores index data. You cannot create or delete the system storage area.

Example

This XML document describes the default SEARCH_DATA storage area:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:storageAreas>
     <search:storageArea>
         <search:name>SEARCH_DATA</search:name>
         <search:description>Default storage area</search:description>
         <search:usage>PARTITION</search:usage>
      </search:storageArea>
   </search:storageAreas>
</search:config>
```

suggContent

The `suggContent` object contains the suggested content configuration settings.

Object Type

Universal

State Properties

None

Supported Operations

```
export
update
```

Administration GUI Page

Search – Suggested Content

XML Description

The `<search:suggContent>` element describes suggested content:

```
<search:suggContent>
    <search:timeout>
    <search:numProviders>
```

Element Descriptions

<search:suggContent>
Describes suggested content parameters. It contains these elements:

```
<search:timeout>
<search:numProviders>
```

<search:timeout>
Time limit, in milliseconds, for Oracle SES to fetch the content. If search result contains suggested content, then the result page is not rendered until the content is available or until the timeout period has expired.

<search:numProviders>
Maximum number of suggested content results (up to 20) to be included with the Oracle SES result list. The results are rendered on a first-come, first-served basis.

Example

This XML document contains the suggested content configuration settings.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:suggContent>
      <search:timeout>2000</search:timeout>
      <search:numProviders>2</search:numProviders>
   </search:suggContent>
</search:config>
```

suggContentProvider

The `suggContentProvider` object contains the information about suggested content providers.

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

Search – Suggested Content - Create, Edit, Delete Provider

XML Description

The `<search:suggContentProviders>` element describes suggested content providers:

```
<search:suggContentProviders>
   <search:suggContentProvider>
      <search:name>
      <search:queryPattern>
      <search:providerUrl>
      <search:xsltStyleSheet>
      <search:securitySettings>
      <!-- securitySettings element can have only one of the following child
elements - none, cookie, or serviceToService>
         <search:none>
         <search:cookie>
            <search:name>
            <search:unauthenticatedUserAction>
            <search:loginUrl>
         <search:serviceToService>
            <search:entityName>
            <search:entityPassword>
            <search:format>
```

Element Descriptions

<search:suggContentProviders>

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

<search:suggContentProvider>

Describes a suggested content provider. It contains these elements:

```
<search:name>
<search:queryPattern>
<search:providerUrl>
<search:xsltStyleSheet>
<search:securitySettings>
```

<search:name>

Name of the suggested content provider.

<search:queryPattern>

Query pattern for the suggested content provider. The query pattern is defined using regular expressions as supported in the Java regular expression API `java.util.regex`. The query pattern must be specified in a CDATA section.

<search:providerUrl>

URL of the suggested content provider.

<search:xsltStyleSheet>

XSLT style sheet that defines rules (for example, the size and style) for transforming XML content from a provider into HTML format. The XSLT style sheet must be specified in a CDATA section.

<search:securitySettings>

Describes how Oracle SES passes end user's authentication information to the suggested content provider. It contains one of the following child elements:

```
<search:none>
<search:cookie>
<search:serviceToService>
```

<search:none>

Describes the option of using no security settings.

<search:cookie>
Describes the option of using security settings by using a cookie to pass user authentication information to the suggested content provider. It contains the following elements.

```
<search:name>
<search:unauthenticatedUserAction>
<search:loginUrl>
```

<search:name>
Name of the cookie.

<search:unauthenticatedUserAction>
Describes what should happen when suggested content is available but the user is not logged in to the content provider or the cookie for the suggested content provider is not available. It can have one of the following values:

- `IGNORE_CONTENT` - Oracle SES returns the result list with no suggested content.
- `DISPLAY_LOGIN_MESSAGE` - Oracle SES returns a message that there is content available from this provider but the user is not logged in. The message also provides a link to log in to that provider. Specify the link for the suggested content provider login in the **<search:loginUrl>** element.

<search:loginUrl>
When `DISPLAY_LOGIN_MESSAGE` value is specified for the **<search:unauthenticatedUserAction>** element, then specify the URL to log in to the suggested content provider in the **<search:loginUrl>** element.

<search:serviceToService>
Describes the option of using security settings by establishing one-way trusted relationship between Oracle SES and the suggested content provider. It contains the following elements.

```
<search:entityName>
<search:entityPassword>
<search:format>
```

<search:entityName>
User name for logging in to the suggested content provider application.

<search:entityPassword>
Password for logging in to the suggested content provider application.

<search:format>
Authentication format for the user logging in to the suggested content provider application.

Example

This XML document contains the definition for a suggested content provider.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:suggContentProviders>
      <search:suggContentProvider>
         <search:name>Provider1</search:name>
         <search:queryPattern><![CDATA[dir (\S+)]]></search:queryPattern>
```

```
<search:providerUrl>http://www.xyz.com:8810/OASearchProvider?query=dir%20john&amp;
p0=dir&amp;p1=john&amp;authType=sso</search:providerUrl>

         <search:xsltStyleSheet>
<![CDATA[<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" indent="no"/>
<xsl:template match="/OneBoxResults">
  <HTML>
   <table border="0" cellpadding="1" cellspacing="0">
    <tr>
      <td>
        <a>
          <xsl:attribute name="href">
            <xsl:value-of select="title/urlLink"/>
          </xsl:attribute>
          <b>
            App HR:<xsl:value-of select="title/urlText"/>
          </b>
        </a>
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
          <tr>
            <td style="vertical-align:middle;width:20px">
              <img alt="Service">
                <xsl:attribute name="src">
                  <xsl:value-of select="IMAGE_SOURCE"/>
                </xsl:attribute>
              </img>
            </td>
            <td style="padding-left:6px;vertical-align:top;">
              <xsl:for-each select="MODULE_RESULT">
                <table width="100%" border="0" cellspacing="0" cellpadding="1">
                  <tr valign="top" align="left">
                    <td colspan="5" align="left">
                      <font size="-1">
                        <b>
                          <xsl:value-of select="Field[@name='DisplayName']"/>
                        </b>
                      </font>
                    </td>
                    <td align="right">
                      <font size="-1">Phone:</font>
                    </td>
                    <td align="left">
                      <font size="-1">
                        <nobr>
                          <b>
                            <xsl:value-of select="Field[@name='WorkTelephone']"/>
                          </b>
                        </nobr>
                      </font>
                    </td>
                    <td style="padding-left:6px">
                    </td>
                    <td align="right">
                      <font size="-1">Email:</font>
                    </td>
                    <td align="left">
                      <font size="-1">
                        <nobr>
```

```
                          <b>
                            <xsl:value-of select="Field[@name='EmailAddress']"/>
                          </b>
                        </nobr>
                      </font>
                    </td>
                    <td align="right">
                      <font size="-1">
                        <nobr>location:</nobr>
                      </font>
                    </td>
                    <td align="left">
                      <font size="-1">
                        <nobr>
                          <b>
                            <xsl:value-of select="Field[@name='DerivedLocale']"/>
                          </b>
                        </nobr>
                      </font>
                    </td>
                    <td>
                    </td>
                  </tr>
                </table>
              </xsl:for-each>
            </td>
          </tr>
        </table>
      </td>
    </tr>
   </table>
  </HTML>
</xsl:template>
</xsl:stylesheet>]]>
         </search:xsltStyleSheet>

         <search:securitySettings>
            <search:cookie>
               <search:name>testcookie1</search:name>
               <search:unauthenticatedUserAction>DISPLAY_LOGIN_
MESSAGE</search:unauthenticatedUserAction>

<search:loginUrl>http://www.xyz.com:8810/OASearchProvider?query=dir%20john&amp;p0=
dir&amp;p1=john&amp;authType=sso</search:loginUrl>
             </search:cookie>
         </search:securitySettings>

      </search:suggContentProvider>
   </search:suggContentProviders>
</search:config>
```

suggestion

The suggestion object is used to create, update, delete, and export suggestions.

Object Type

Creatable

Object Key

name, classification, language

Supported Operations

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

Administration GUI Page

None

XML Description

The <search:suggestions> element describes suggestions:

```
<search:suggestions>
   <search:suggestion>
      <search:name>
      <search:classification>
      <search:language>
      <search:weight>
```

Element Descriptions

<search:suggestions>
Contains one or more <search:suggestion> element.

<search:suggestion>
Contains these elements:

```
<search:name>
<search:classification>
<search:language>
<search:weight>
```

<search:name>
The suggestion keyword.

<search:classification>
The classification for the suggestion keyword.

<search:language>
The language for which this suggestion keyword should be displayed. It is pecified using a two letter code. The language codes are not case sensitive. See Table 2–3, " Languages Supported by the Crawler". The value `any` can also be specified for the language, denoting that the suggestion is language independent.

<search:weight>
Specify weight for the suggestion keyword, based on which Oracle SES retrieves and sorts suggestions, with the highest weighted suggestions being displayed first in the search results.

Example

This XML document contains the definition for the suggestion keyword `ses`.

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:suggestions>
      <search:suggestion>
         <search:name>ses</search:name>
         <search:classification>OracleGeneric</search:classification>
         <search:language>en</search:language>
         <search:weight>10</search:weight>
      </search:suggestion>
   </search:suggestions>
</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/technetwork/search/oses/overview/index.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(`*term*`)`	about(dogs)
and	*term* `AND` *term*	dog and cat
near	*term*`;`*term*	dog ; cat
or	*term* `OR` *term*	dog or cat
phrase	*phrase*	dog sled
stem	*$term*	$dog
thesaurus	`{BT | NT | SYN}` (*term*)	SYN(dog)
within	*term* `WITHIN` *term*	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="11.2.2.2.0"
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>
```

tagging

The `tagging` object contains the tagging related configuration settings.

Object Type

Universal

State Properties

Property	Value
status	ACTIVE INACTIVE

Supported Operations

```
activate
deactivate
export
getState
update
```

Administration GUI Page

Search - Tagging

XML Description

The `<search:tagging>` element describes the tagging configurations:

```
<search:tagging>
   <search:maxTagPerDoc>
   <search:maxTagPerSession>
   <search:tagCleanupInterval>
   <search:authorizationMode>
```

Element Descriptions

<search:tagging>
Contains the following elements:

```
<search:maxTagPerDoc>
<search:maxTagPerSession>
<search:tagCleanupInterval>
<search:authorizationMode>
```

<search:maxTagPerDoc>
Maximum number of tags that can be assigned to a document (not specific to a user). The default value is 100.

<search:maxTagPerSession>
Maximum number of tags that can be added in a session. The default value is 100.

<search:tagCleanupInterval>
Number of days for which any tag should be available in the query application, even if it is not being used. When the number of days specified in tagCleanupInterval elapse,

the tags that are unused for the specified number of days are removed from Oracle SES. The default value is 30.

<search:authorizationMode>
Specify one of the following authorization modes:

Tagging Mode	Description
`loggedInUsers`	Tagging is enabled only for the users who are logged-in. This is default.
`allUsers`	Tagging is enabled for all the users (anonymous tagging).
`authorizedPrincipals`	Tagging is enabled only for specific users having tagging privilege.

Example

This XML document defines the tagging configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:tagging>
      <search:maxTagPerDoc>100</search:maxTagPerDoc>
      <search:maxTagPerSession>100</search:maxTagPerSession>
      <search:tagCleanupInterval>30</search:tagCleanupInterval>
      <search:authorizationMode><search:loggedInUsers/></search:authorizationMode>
   </search:tagging>
</search:config>
```

tag

The tag object can be used to upload tags in bulk in Oracle SES.

Object Type

Creatable

Object Key

`name, docUrl, owner`

Object Key Command Syntax

```
--NAME=object_name
-n object_name
--DOC_URL=url
--OWNER=owner
```

State Properties

None

Supported Operations

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

Administration GUI Page

None

XML Description

The `<search:tags>` element describes the tags for bulk upload:

```
<search:tags>
   <search:tag>
      <search:name>
      <search:docURL>
      <search:owner>
```

Element Descriptions

<search:tags>
Contains one or more `<search:tag>` elements.

<search:tag>
Contains information for each tag. It contains the following elements:

```
<search:name>
<search:docURL>
<search:owner>
```

<search:name>
Name of the tag.

<search:docURL>
URL of the document that needs to be tagged.

<search:owner>
Owner of the document.

Example

This XML document contains the tags for bulk upload:

```
<?xml version="1.0" encoding="UTF-8"?>
<search:config productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:tags>
    <search:tag>
      <search:name>oses</search:name>
      <search:docUrl>http://www.oracle.com/xyz.html</search:docUrl>
      <search:owner>abc@oracle.com</search:owner>
    </search:tag>
  </search:tags>
</search:config>
```

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="11.2.2.2.0"
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>
```

3 searchadmin Commands

This chapter describes all the `searchadmin` commands of Oracle SES.

Alphabetic List of searchadmin Commands

A C D E G S U

A

`activate clustering`
`activate clusterTree`
`activate identityPlugin`
`activate indexOptimizer`
`activate partitionConfig`
`activate queryUIFacets`
`activate resultList`
`activate schedule`
`activate singleSignOnSetting`
`activate skinBundle`
`activate suggContentProvider`
`activate tagging`

C

`create altWord`
`create boostedUrl`
`create classification`
`create clusterTree`
`create docServiceInstance`
`create docServiceManager`
`create docServicePipeline`
`create facetTree`
`create identityPlugin`
`create indexProfile`
`create lexer`
`create proxyLogin`
`create schedule`
`create searchAttr`
`create skinBundle`
`create source`
`create sourceGroup`
`create sourceType`
`create storageArea`
`create suggContentProvider`
`create suggestion`
`create suggLink`
`create tag`

create thesaurus
createAll altWord
createAll authorizedPrincipal
createAll classification
createAll clusterTree
createAll docServiceInstance
createAll docServiceManager
createAll docServicePipeline
createAll facetTree
createAll identityPlugin
createAll indexProfile
createAll lexer
createAll proxyLogin
createAll schedule
createAll searchAttr
createAll source
createAll sourceGroup
createAll sourceType
createAll storageArea
createAll suggContentProvider
createAll suggestion
createAll suggLink

D

deactivate clustering
deactivate clusterTree
deactivate identityPlugin
deactivate indexOptimizer
deactivate queryUIFacets
deactivate resultList
deactivate schedule
deactivate singleSignOnSetting
deactivate skinBundle
deactivate suggContentProvider
deactivate tagging
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 singleSignOnSetting
getAllStates skinBundle
getAllStates suggContentProvider
getState autoSuggestion
getState clustering
getState clusterTree
getState identityPlugin
getState index

```
getState indexOptimizer
getState queryUIFacets
getState resultList
getState schedule
getState singleSignOnSetting
getState skinBundle
getState suggContentProvider
getState tagging
getStateList clusterTree
getStateList identityPlugin
getStateList schedule
getStateList singleSignOnSetting
getStateList skinBundle
getStateList suggContentProvider
```

S

```
start autoSuggestion
start indexOptimizer
start schedule
stop indexOptimizer
stop schedule
```

U

```
update altWord
update autoSuggestion
update boostedUrl
update classification
update classificationMappings
update clustering
update clusterTree
update crawlerSettings
update docServiceInstance
update docServicePipeline
update facetTree
update globalBoundaryRules
update globalDocumentTypes
update index
update indexOptimizer
update indexProfile
update languageBasedTokenization
update lexer
update partitionConfig
update proxy
update proxyLogin
update queryConfig
update queryUIConfig
update queryUIFacets
update relevanceRanking
update resultList
update schedule
update searchAttr
update singleSignOnSetting
update skinBundle
update source
update sourceGroup
update sourceType
update storageArea
update suggContent
update suggContentProvider
update suggestion
update suggLink
update tagging
update thesaurus
updateAll altWord
updateAll boostedUrl
updateAll classification
updateAll clusterTree
updateAll docServiceInstance
updateAll docServicePipeline
updateAll facetTree
updateAll indexProfile
updateAll lexer
updateAll proxyLogin
updateAll schedule
updateAll searchAttr
updateAll singleSignOnSetting
updateAll sourceGroup
updateAll sourceType
updateAll storageArea
updateAll suggContentProvider
updateAll suggestion
updateAll suggLink
```

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 2-65.

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 partitionConfig

Activates partitionConfig.

Syntax

```
activate partitionConfig
```

Example

This example activates partitionConfig:

```
SES>activate partitionConfig

The object "partitionConfig" was successfully activated.
```

activate queryUIFacets

Activates queryUIFacets.

Syntax

```
activate queryUIFacets
```

Example

This example activates queryUIFacets:

```
SES>activate queryUIFacets

The object "queryUIFacets" 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 2-125.

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 singleSignOnSetting

Activates a Single Sign-On setting.

Syntax

```
activate singleSignOnSetting --NAME=object_name
```

or

```
activate singleSignOnSetting -n object_name]
```

Parameters

object_name
Content of a `<search:name>` element in the XML document.

Example

This example activates a Single Sign-On setting named `OSSO`:

```
SES>activate singleSignOnSetting --NAME="OSSO"

The object "[name=OSSO]" was successfully activated.
```

activate skinBundle

Activates a skin bundle.

See Also

"Search Interface Customization: Skin Bundles" on page 2-11

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 2-135.

list_filename
Path to the text file that lists the files in the skin bundle. See the Notes for "create skinBundle" on page 3-31.

Example

This example activates a skin bundle named `acme`:

```
SES>activate skinBundle --NAME=acme

The object "[name=acme]" was successfully activated
```

activate suggContentProvider

Activates a suggested content provider.

Syntax

```
activate suggContentProvider --NAME=object_name
```

or

```
activate suggContentProvider -n object_name]
```

Parameters

object_name
Content of a `<search:name>` element in the XML document.

Example

This example activates a suggested content provider named `Provider1`:

```
SES>activate suggContentProvider --NAME="Provider1"

The object "[name=Provider1]" was successfully activated.
```

activate tagging

Activates tagging configuration.

Syntax

```
activate tagging
```

Example

This example activates tagging configuration:

```
SES>activate tagging

The object "tagging" 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 2-20.

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 boostedUrl

Creates a boostedUrl object from an XML description.

See Also

createAll authorizedPrincipal

Syntax

```
create boostedUrl --DOC_URL=url --QUERY=query_term
```

Parameters

url
The URL of a document being boosted.

query_term
The exact query term that returns the boosted URL.

Example

This example boosts a URL for queries matching the term "indexing":

```
SES>create boostedUrl --DOC_URL=http://example.com//doctools/b32440/xref_foot_
in.htm --QUERY=indexing --INPUT_FILE=boost.xml

The object "[query=indexing, docUrl=http://example.com/doctools/b32440/xref_foot_
in.htm]" was successfully created.
```

create classification

Creates classification from an XML description.

See Also

`createAll classification`

Syntax

```
create classification --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create classification -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 "classification" on page 2-29.

Example

This example creates a classfication `myClassification` defined in `classification.xml`.

```
SES>create classification --NAME="myClassification" --INPUT_
FILE=classification.xml

The object "[name=myClassification]" 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 2-36.

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 docServiceInstance

Creates a document service instance from an XML description.

See Also

createAll docServiceInstance

Syntax

```
create docServiceInstance --NAME=object_name --INPUT_FILE=xml_filename [--ENCRYPT_
KEY=key]
```

or

```
create docServiceInstance -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 "docServiceInstance" on page 2-45.

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 a service instance from the description in docserviceinstance.xml:

```
SES>create docServiceInstance --NAME="Default summarizer service instance"
--INPUT_FILE=docserviceinstance.xml --ENCRYPT_KEY=key2encrypt

The object "[name=Default summarizer service instance]" was successfully created.
```

create docServiceManager

Creates a document service manager from an XML description.

See Also

createAll docServiceManager

Syntax

Check this syntax -- doesn't match example

```
create docServiceManager --NAME=object_name --INPUT_FILE=xml_filename]
```

or

```
create docServiceManager -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 "docServiceManager" on page 2-48.

Example

```
SES>create docServiceManager --JAR_FILE=extractor/extractor.jar --MANAGER_
CLASS=oracle.search.plugin.doc.extractor.DocumentSummarizerManager --INPUT_
FILE=docservicemanager.xml

The object "[jarFilePath=extractor/extractor.jar,
managerClassName=oracle.search.plugin.doc.extractor.DocumentSummarizerManager]"
was successfully created.
```

create docServicePipeline

Creates a document service pipeline from an XML description.

See Also

createAll docServicePipeline

Syntax

```
create docServicePipeline --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create docServicePipeline -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 "docServicePipeline" on page 2-51.

Example

This example creates a document service pipeline:

```
SES>create docServicePipeline --NAME=Default pipeline --INPUT_
FILE=docservicepipeline.xml

The object "[name=Default pipeline]" was successfully created.
```

create facetTree

Creates a facet tree from an XML description.

See Also

createAll facetTree

Syntax

```
create facetTree --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create facetTree -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 "facetTree" on page 2-54.

Example

This example creates a facet tree named Country:

```
SES>create facetTree --FACETNAME=Country --INPUT_FILE=facettree.xml

The object "[facetName=Country]" 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 2-65.

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 indexProfile

Creates an index profile from an XML description.

See Also

createAll indexProfile

Syntax

```
create indexProfile --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create indexProfile -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 "indexProfile" on page 2-74.

Example

This example creates an index profile named My Index Profile:

```
SES>create indexProfile --NAME="My IndexProfile" --INPUT_FILE=indexprofile.xml

The object "[name=My Index Profile]" was successfully created.
```

create lexer

Creates a lexer from an XML document.

See Also

createAll lexer

Syntax

```
create lexer --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create lexer -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 "lexer" on page 2-79.

Example

This example creates a Swedish lexer:

```
SES>create lexer --NAME=SwedishLexer --INPUT_FILE=lexer.xml

The object "[name=SwedishLexer]" 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 2-94.

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 3-32.

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 2-125.

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 2-129.

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 2-11

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 2-135.

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 2-137.

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 2-165.

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 2-167.

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. Note that this command does not create a new tablespace, but uses an existing tablespace for creating `storageArea` object.

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 2-171.

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 suggContentProvider

Creates a suggested content provider from an XML description.

See Also

createAll suggContentProvider

Syntax

```
create suggContentProvider --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create suggContentProvider -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 "suggContentProvider" on page 2-174.

Example

This example creates a suggested content provider named `Provider1`:

```
SES>create suggContentProvider --NAME=Provider1 --INPUT_FILE=provider1.xml

The object "[name=Provider1]" was successfully created.
```

create suggestion

Creates a suggestion from an XML description.

See Also

`createAll suggestion`

Syntax

```
create suggestion --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create suggestion -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 "suggestion" on page 2-179.

Example

This example creates a suggestion `mySuggestion` defined in `suggestion.xml`.

```
SES>create suggestion --NAME="mySuggestion" --INPUT_FILE=suggestion.xml

The object "[name=mySuggestion]" 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 2-181.

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 tag

Creates a tag from an XML description.

Syntax

```
create tag --NAME=object_name --INPUT_FILE=xml_filename
```

or

```
create tag -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 "tag" on page 2-185.

Example

This example creates the `OSES` tag defined in tag.xml:

```
SES>create tag --NAME=OSES --INPUT_FILE=tag.xml

The object "[name=OSES]" 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 2-187.

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 2-20.

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 boostedUrl

Creates all the boostedUrl objects described in an XML document.

See Also

create boostedUrl

Syntax

```
createAll boostedUrl --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll boostedUrl -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "boostedUrl" on page 2-27.

action
Action to take when an object already exists:

- error: The 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 one boostedUrl defined in boost.xml and ignores the existing object description.

```
SES>createAll boostedUrl --INPUT_FILE=boost.xml --DUPE_METHOD=ignore

createAll operation succeeded for type "boostedUrl".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_IGNORED
```

createAll authorizedPrincipal

Creates all the authorizedPrincipal objects described in an XML document.

Syntax

```
createAll authorizedPrincipal --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll authorizedPrincipal -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "authorizedPrincipal" on page 2-22.

action
Action to take when an object already exists:

- `error`: The 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 one `authorizedPrincipal` defined in authorizedPrincipals.xml and ignores the existing object description.

```
SES>createAll authorizedPrincipal --INPUT_FILE=authorizedPrincipals.xml --DUPE_
METHOD=ignore

createAll operation succeeded for type "authorizedPrincipal".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_IGNORED
```

createAll classification

Creates all the classifications described in an XML file.

See Also

`create classification`

Syntax

```
createAll classification --INPUT_FILE=xml_filename [--DUPE_METHOD=action
[--IGNORE_INVALID_STATE=state]]
```

or

```
createAll classification -i xml_filename [-d action [-s state]]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "classification" on page 2-29.

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 classifications described in classifications.xml. The second classification already exists.

```
SES>createAll classification --INPUT_FILE=classifications.xml --DUPE_
METHOD=overwrite

createAll operation succeeded for type "classification".

  1 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 2-36.

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 docServiceInstance

Creates all document service instances described in an XML file.

See Also

create docServiceInstance

Syntax

```
createAll docServiceInstance --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll docServiceInstance -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "docServiceInstance" on page 2-45.

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 one new document service instance and overwrites an exiting one:

```
SES>createAll docServiceInstance --INPUT_FILE=docserviceinstance.xml --DUPE_
METHOD=overwrite --ENCRYPT_KEY=key2encrypt

createAll operation succeeded for type "docServiceInstance".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll docServiceManager

Creates all document service managers described in an XML file.

See Also

create docServiceManager

Syntax

```
createAll docServiceManager --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll docServiceManager -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "docServiceManager" on page 2-48.

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.

Example

The following example shows the resulting error when `--DUPE_METHOD=error` and an object described in the XML file already exists. No objects are created.

```
SES>createAll docServiceManager --INPUT_FILE=docservicemanager.xml --DUPE_
METHOD=error

EQA-11001: The object with key "[jarFilePath=extractor/extractor.jar,
managerClassName=oracle.search.plugin.doc.extractor.DocumentSummarizerManager]"
and type "docServiceManager" already exists.
```

createAll docServicePipeline

Creates all document service pipelines described in an XML document.

See Also

create docServicePipeline

Syntax

```
createAll docServicePipeline --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll docServicePipeline -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "docServicePipeline" on page 2-51.

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 new document service pipeline and replaces an existing one:

```
SES>createAll docServicePipeline --INPUT_FILE=docservicepipeline.xml --DUPE_
METHOD=overwrite

createAll operation succeeded for type "docServicePipeline".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll facetTree

Creates all facet trees described in an XML document.

See Also

create facetTree

Syntax

```
createAll facetTree --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll facetTree -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "facetTree" on page 2-54.

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.

Example

This example attempts to create two facet trees. One facet tree is created successfully, and the other attempt is ignored because the object already exists.

```
SES>createAll facetTree --INPUT_FILE=facettree.xml --DUPE_METHOD=ignore

createAll operation succeeded for type "facetTree".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_IGNORED
```

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 2-65.

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 indexProfile

Creates all index profiles described in an XML document

See Also

create indexProfile

Syntax

```
createAll indexProfile --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll indexProfile -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "indexProfile" on page 2-74.

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

```
SES>createAll indexProfile --INPUT_FILE=indexprofile.xml --DUPE_METHOD=overwrite

createAll operation succeeded for type "indexProfile".

  2 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll lexer

Creates all lexers described in an XML document.

See Also

create lexer

Syntax

```
createAll lexer --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll lexer -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "lexer" on page 2-79.

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 overwrites three lexers:

```
SES>createAll lexer --INPUT_FILE=lexer.xml --DUPE_METHOD=overwrite

createAll operation succeeded for type "lexer".

  3 object(s) with status DUPLICATE_OVERWRITTEN
```

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 2-94.

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 2-125.

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 2-129.

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 searchattrx.xml. A fourth object already exists.

```
SES>createAll searchAttr --INPUT_FILE=searchattrs.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 2-137.

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 2-165.

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 2-167.

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 2-36.

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 suggContentProvider

Creates all the suggested content providers described in an XML file.

See Also

create suggContentProvider

Syntax

```
createAll suggContentProvider --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll suggContentProvider -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "suggContentProvider" on page 2-174.

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 two suggested content providers from the descriptions in `suggcontentproviders.xml`. A third object already exists.

```
SES>createAll suggContentProvider --INPUT_FILE=suggcontentproviders.xml --DUPE_
METHOD=overwrite

createAll operation succeeded for type "suggContentProvider".

  2 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_OVERWRITTEN
```

createAll suggestion

Creates all the suggestions described in an XML file.

See Also

create suggestion

Syntax

```
createAll suggestion --INPUT_FILE=xml_filename [--DUPE_METHOD=action [--IGNORE_
INVALID_STATE=state]]
```

or

```
createAll suggestion -i xml_filename [-d action [-s state]]
```

Parameters

xml_filename

Path to the XML document that contains the object descriptions. See "suggestion" on page 2-179.

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 suggestions described in suggestions.xml. The second suggestion already exists.

```
SES>createAll suggestion --INPUT_FILE=suggestions.xml --DUPE_METHOD=overwrite

createAll operation succeeded for type "suggestion".

  1 object(s) with status CREATE_SUCCEEDED
  1 object(s) with status DUPLICATE_OVERWRITTEN
```

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 2-181.

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
```

createAll tag

Creates all the tags described in an XML file.

See Also

create tag

Syntax

```
createAll tag --INPUT_FILE=xml_filename [--DUPE_METHOD=action]
```

or

```
createAll tag -i xml_filename [-d action]
```

Parameters

xml_filename
Path to the XML document that contains the object descriptions. See "tag" on page 2-185.

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 tags from the descriptions in tags.xml. A third object already exists.

```
SES>createAll tag --INPUT_FILE=tags.xml --DUPE_METHOD=overwrite

createAll operation succeeded for type "tag".

  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 queryUIFacets

Deactivates queryUIFacets.

Syntax

```
deactivate queryUIFacets
```

Example

This example deactivates queryUIFacets:

```
SES>deactivate queryUIFacets

The object "queryUIFacets" 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 singleSignOnSetting

Deactivates a Single Sign-On setting.

Syntax

```
deactivate singleSignOnSetting --NAME=object_name
```

or

```
deactivate singleSignOnSetting -n object_name
```

Parameters

object_name
Content of a `<search:name>` element in the XML document.

Example

This example deactivates a Single Sign-On setting named *OSSO*:

```
SES>deactivate singleSignOnSetting --NAME="OSSO"

The object "[name=OSSO]" was successfully deactivated.
```

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 deactivates a skin bundle named *holiday*:

```
SES>deactivate skinBundle --NAME=holiday

The object "[name=holiday]" was successfully deactivated.
```

deactivate suggContentProvider

Deactivates a suggested content provider.

Syntax

```
deactivate suggContentProvider --NAME=object_name
```

or

```
deactivate suggContentProvider -n object_name
```

Parameters

object_name
Content of a `<search:name>` element in the XML document.

Example

This example deactivates a suggested content provider named *Provider1*:

```
SES>deactivate suggContentProvider --NAME=Provider1

The object "[name=Provider1]" was successfully deactivated.
```

deactivate tagging

Deactivates the tagging configuration.

Syntax

```
deactivate tagging
```

Example

This example deactivates the tagging configuration:

```
SES>deactivate tagging

The object "tagging" 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:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
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:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

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:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

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:

```
altWord
boostedUrl
classification
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
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="11.2.2.2.0"
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:

```
autoSuggestion
classificationMappings
clustering
crawlerSettings
index
indexOptimizer
languageBasedTokenization
partitionConfig
proxy
queryConfig
queryUIConfig
queryUIFacets
queryUISourceGroups
relevanceRanking
resultList
tagging
```

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="11.2.2.2.0"
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:

```
altWord
boostedUrl
classification
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

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="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:suggLinks>
      <search:suggLink>
         <search:keyword>database</search:keyword>
         <search:linkUrl>
          http://www.oracle.com/technetwork/database/enterprise-edition/index.html
         </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/technetwork/search/oses/overview/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="11.2.2.2.0"
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:

```
altWord
boostedUrl
classification
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

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="11.2.2.2.0"
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="11.2.2.2.0"
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
authorizedPrincipal
boosteUrl
classification
clusterTree
docServiceInstance
docServiceManager
docServicePipeline
facetTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
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

11.2.2.2.0
```

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="11.2.2.2.0"
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="11.2.2.2.0"
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="11.2.2.2.0"
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 singleSignOnSetting

Returns the current state of all Single Sign-On settings as an XML document.

See Also

getState singleSignOnSetting
getStateList singleSignOnSetting

Syntax

```
getAllStates singleSignOnSetting [--OUTPUT_FILE=output_file]
```

or

```
getAllStates singleSignOnSetting [-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 getAllStates command creates a separate file for each one and appends the host name and port number to the base name.

Example

This example returns the current state of all the Single Sign-On settings:

```
SES>getAllStates singleSignOnSetting

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>singleSignOnSetting</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>OSSO</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>singleSignOnSetting</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>OMA</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>singleSignOnSetting</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>WNA</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>
```

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="11.2.2.2.0"
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>
```

getAllStates suggContentProvider

Returns the current state of all the suggested content providers as an XML document.

See Also

getState suggContentProvider
getStateList schedule

Syntax

```
getAllStates suggContentProvider [--OUTPUT_FILE=output_file] [--PROPERTY_
LIST=status]
```

or

```
getAllStates suggContentProvider [-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 `getAllStates` 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 all the suggested content providers.

```
SES>getAllStates suggContentProvider

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>suggContentProvider</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>Provider1</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 autoSuggestion

Returns the current state of auto suggestion configuration as an XML document.

Syntax

```
getState autoSuggestion [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState autoSuggestion [-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 gets the current state of auto suggestion configuration:

```
SES>getState autoSuggestion

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>autoSuggestion</search:objectType>
         <search:stateProperties>
            <search:stateProperty>
               <search:propertyName>filterStatus</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 configuration that is currently active:

```
SES>getState clustering

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
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="11.2.2.2.0"
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>
```

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="11.2.2.2.0"
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>
```

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="11.2.2.2.0"
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 2-71 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="11.2.2.2.0"
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 partitionConfig

Returns the current state of partitionConfig configuration as an XML document.

Syntax

```
getState partitionConfig [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState partitionConfig [-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 gets the current state of partitionConfig configuration:

```
SES>getState partitionConfig

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>partitionConfig</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 queryUIFacets

Returns the current state of queryUIFacets configuration as an XML document.

Syntax

```
getState queryUIFacets [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState queryUIFacets [-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 gets the current state of queryUIFacets configuration:

```
SES>getState queryUIFacets

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>queryUIFacets</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 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="11.2.2.2.0"
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="11.2.2.2.0" 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: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 singleSignOnSetting

Returns the current state of a Single Sign-On setting as an XML document.

See Also

`getAllStates singleSignOnSetting`
`getStateList singleSignOnSetting`

Syntax

```
getState singleSignOnSetting --NAME=object_name [--OUTPUT_FILE=output_file]
```

or

```
getState singleSignOnSetting -n object_name [-o output_file]
```

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 WNA Single Sign-On setting is currently inactive.

```
SES>getState singleSignOnSetting -n WNA

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>singleSignOnSetting</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>WNA</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 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="11.2.2.2.0"
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 suggContentProvider

Returns the current state of a suggested content provider as an XML document.

See Also

getAllStates suggContentProvider
getStateList skinBundle

Syntax

```
getState suggContentProvider --NAME=object_name [--OUTPUT_FILE=output_file]
[--PROPERTY_LIST=status]
```

or

```
getState suggContentProvider -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 suggested content provider *Provider1* is currently active.

```
SES>getState suggContentProvider -n Provider1

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>suggContentProvider</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>Provider1</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 tagging

Returns the current state of the tagging configuration as an XML document.

Syntax

```
getState tagging [--OUTPUT_FILE=output_file] [--PROPERTY_LIST=status]
```

or

```
getState tagging [-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 that the tagging configuration is currently active:

```
SES>getState tagging

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>tagging</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>
```

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="11.2.2.2.0"
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>
```

getStateList identityPlugin

Returns the current state of a list of identity plugin objects.

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="11.2.2.2.0"
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: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="11.2.2.2.0"
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>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 singleSignOnSetting

Returns the current state of list of Single Sign-On settings.

See Also

getAllStates singleSignOnSetting
getState singleSignOnSetting

Syntax

```
getStateList singleSignOnSetting --KEYS_FILE=key_filename [--OUTPUT_FILE=output_
file]
```

or

```
getStateList singleSignOnSetting -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="OAM"
--NAME="OSSO"
```

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 stores the current state of all Single Sign-On settings listed in singlesignonsettings.lst in a file named singlesignonsettingsstate.xml.

```
SES>getStateList singleSignOnSetting --KEYS_FILE=singlesignonsettings.lst
--OUTPUT_FILE=singlesignonsettingsstate.xml

The getStateList operation succeeded.
```

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.
```

getStateList suggContentProvider

Returns the current state of a list of suggested content provider objects.

See Also

getState suggContentProvider
getAllStates suggContentProvider

Syntax

```
getStateList suggContentProvider --KEYS_FILE=key_filename [--OUTPUT_FILE=output_
file]
```

or

```
getStateList suggContentProvider -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="Provider1"
```

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 status for the suggested content providers listed in `suggcontentproviders.lst`:

```
SES>getStateList suggContentProvider --KEYS_FILE=suggcontentproviders.lst

<?xml version="1.0" encoding="UTF-8"?>
<search:state productVersion="11.2.2.2.0"
xmlns:search="http://xmlns.oracle.com/search">
   <search:objectStates>
      <search:objectState>
         <search:objectType>suggContentProvider</search:objectType>
         <search:objectKey>
            <search:keyPairs>
               <search:keyPair>
                  <search:name>name</search:name>
                  <search:value>Provider1</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>
```

start autoSuggestion

Starts jobs for applying filter expression specified in `<search:filterExpression>` to suggestion keywords that have already been populated from queries.

Syntax

```
start autoSuggestion --ACTION=filter
```

Example

This example starts job that removes existing suggestion keywords populated from queries that match the filter expression:

```
SES>start autoSuggestion --ACTION=filter

The "start" operation succeeded for "autoSuggestion".
```

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]".
```

stop indexOptimizer

Stops index optimization.

Syntax

```
stop indexOptimizer
```

Example

This example stops the index optimizer:

```
SES>stop indexOptimizer

The "stop" operation succeeded for "indexOptimizer".
```

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 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 2-20.

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 autoSuggestion

Sets one or more auto suggestion properties from an XML description.

Syntax

```
update autoSuggestion --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update autoSuggestion -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 2-33.

Example

This example updates the clustering properties:

```
SES>update autoSuggestion --INPUT_FILE=autoSuggestion.xml --UPDATE_
METHOD=overwrite

The object "autoSuggestion" was successfully updated.
```

update boostedUrl

Changes the properties of a boosted URL from an XML document.

See Also

`updateAll boostedUrl`

Syntax

```
update boostedUrl --DOC_URL=url --QUERY=query_term --INPUT_FILE=xml_filename
--UPDATE_METHOD=method
```

or

```
update boostedUrl --DOC_URL=url --QUERY=query_term -i xml_filename -a method
```

Parameters

url
The URL of a document being boosted.

query_term
The exact query term that returns the boosted URL.

xml_filename
Path to the XML document that configures the object. See "boostedUrl" on page 2-27.

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 definition of a boosted URL for the term "indexing":

```
SES>update boostedUrl --DOC_URL="http://example.com/doctools/b32440/xref_foot_
in.htm" --QUERY="indexing" --UPDATE_METHOD=overwrite --INPUT_FILE=boost.xml

The object "[query=indexing, docUrl=http://example.com/doctools/b32440/xref_foot_
in.htm]" was successfully updated
```

update classification

Updates a classification.

See Also

updateAll classification

Syntax

```
update classification --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update classification -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 "classification" on page 2-29.

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 classification_1:

```
SES>update classification --NAME=classification_1 --INPUT_FILE=classification_
update.xml --UPDATE_METHOD=overwrite

The object "[name=classification_1]" was successfully updated.
```

update classificationMappings

Sets one or more classificationMappings properties from an XML description.

Syntax

```
update classificationMappings --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update classificationMappings -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 "classificationMappings" on page 2-31.

Example

This example updates the classificationMappings properties:

```
SES>update classificationMappings --INPUT_FILE=classificationMappings.xml
--UPDATE_METHOD=overwrite

The object "classificationMappings" 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 2-33.

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 2-36.

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 2-40.

Example

This example updates the crawler settings:

```
SES>update crawlerSettings --INPUT_FILE=crawler.xml --UPDATE_METHOD=overwrite

The object "crawlerSettings" was successfully updated.
```

update docServiceInstance

Sets one or more parameters of a document service instance from an XML description.

See Also

updateAll docServiceInstance

Syntax

```
update docServiceInstance --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method [--ENCRYPT_KEY=key]
```

or

```
update docServiceInstance -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 "docServiceInstance" on page 2-45.

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 updates a the default summarizer document service instance:

```
SES>update docServiceInstance --NAME="Default summarizer service instance"
--UPDATE_METHOD=overwrite --INPUT_FILE=docserviceinstance.xml --ENCRYPT_
KEY=key2encrypt

The object "[name=Default summarizer service instance]" was successfully updated.
```

update docServicePipeline

Sets one or more parameters of a document service pipeline from an XML description.

See Also

updateAll docServicePipeline

Syntax

```
update docServicePipeline --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update docServicePipeline -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 "docServicePipeline" on page 2-51.

Example

This example updates the default document service pipeline:

```
SES>update docServicePipeline --NAME=Default pipeline --UPDATE_METHOD=add --INPUT_
FILE=docservicepipeline.xml

The object "[name=Default pipeline]" was successfully updated.
```

update facetTree

Updates a facet tree from an XML description.

See Also

updateAll facetTree

Syntax

```
update facetTree --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update facetTree -a method -i xml_filename
```

Parameters

object_name
Content of a `<search:name>` element in the XML document.

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 defines the object. See "facetTree" on page 2-54.

Example

This example updates a facet tree named Country:

```
SES>update facetTree --FACETNAME=Country --UPDATE_METHOD=add --INPUT_
FILE=facettree.xml

The object "[facetName=Country]" was successfully updated.
```

update globalBoundaryRules

Sets one or more global boundary rules from an XML document.

Syntax

```
update globalBoundaryRules --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update globalBoundaryRules -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 "globalBoundaryRules" on page 2-60.

Example

This example updates the global boundary rules:

```
SES>update globalBoundaryRules -a overwrite --INPUT_FILE=globalboundaryrules.xml

The object "globalBoundaryRules" was successfully updated.
```

update globalDocumentTypes

Sets one or more parameters of a global document type from an XML document.

Syntax

```
update globalDocumentTypes --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update globalDocumentTypes -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 "globalDocumentTypes" on page 2-63.

Example

This example updates the global document types:

```
SES>update globalDocumentTypes -a overwrite --INPUT_FILE=globaldocumenttypes.xml

The object "globalDocumentTypes" 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 2-69.

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 2-71.

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 indexProfile

Sets one or more parameters of an index profile from an XML document.

See Also

updateAll indexProfile

Syntax

```
update indexProfile --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update indexProfile -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 "indexProfile" on page 2-74.

Example

This example updates My Index Profile:

```
SES>update indexProfile --NAME="My IndexProfile" --UPDATE_METHOD=add --INPUT_
FILE=indexprofile.xml

The object "[name=My Index Profile]" was successfully updated.
```

update languageBasedTokenization

Sets one or more parameters for language-based tokenization from an XML document.

Syntax

```
update languageBasedTokenization --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update languageBasedTokenization -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 "languageBasedTokenization" on page 2-77.

Example

This example updates the mapping rules for language-based tokenization.

```
SES>update languageBasedTokenization --UPDATE_METHOD=add --INPUT_
FILE=languagebasedtokenization.xml

The object "languageBasedTokenization" was successfully updated.
```

update lexer

Sets one or more parameters of a lexer from an XML document.

See Also

updateAll lexer

Syntax

```
update lexer --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update lexer -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 "lexer" on page 2-79.

Example

This example updates the default German lexer:

```
SES>update lexer --NAME=OracleDefaultGermanLexer --UPDATE_METHOD=add --INPUT_
FILE=lexer.xml

The object "[name=OracleDefaultGermanLexer]" 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 2-87.

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 proxy

Sets HTTP proxy server configuration parameters from an XML description.

Syntax

```
update proxy --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update proxy -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 "proxy" on page 2-92.

Example

This example updates the HTTP proxy server settings:

```
SES>update proxy --INPUT_FILE=proxy.xml

The object "proxy" 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 2-94.

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 2-96.

Example

This example updates the query settings:

```
SES>update queryConfig --INPUT_FILE=query.xml

The object "queryConfig" was successfully updated.
```

update queryUIConfig

Sets one or more queryUIConfig properties from an XML description.

Syntax

```
update queryUIConfig --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update queryUIConfig -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 "queryUIConfig" on page 2-102.

Example

This example updates the queryUIConfig properties:

```
SES>update queryUIConfig --INPUT_FILE=queryUIConfig.xml --UPDATE_METHOD=overwrite

The object "queryUIConfig" was successfully updated.
```

update queryUIFacets

Sets one or more queryUIFacets properties from an XML description.

Syntax

```
update queryUIFacets --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update queryUIFacets -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 "queryUIFacets" on page 2-108.

Example

This example updates the queryUIFacets properties:

```
SES>update queryUIFacets --INPUT_FILE=queryUIFacets.xml --UPDATE_METHOD=overwrite

The object "queryUIFacets" was successfully updated.
```

update queryUISourceGroups

Sets one or more queryUISourceGroups properties from an XML description.

Syntax

```
update queryUISourceGroups --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update queryUISourceGroups -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 "queryUISourceGroups" on page 2-110.

Example

This example updates the queryUISourceGroups properties:

```
SES>update queryUISourceGroups --INPUT_FILE=queryUIFacets.xml --UPDATE_
METHOD=overwrite

The object "queryUISourceGroups" was successfully updated.
```

update relevanceRanking

Sets one or more relevance ranking parameters from an XML description.

Syntax

```
update relevanceRanking --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update relevanceRanking -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 "relevanceRanking" on page 2-115.

Example

This example updates the ranking attributes:

```
SES>update relevanceRanking -a remove --INPUT_FILE=relevance.xml

The object "relevanceRanking" 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 2-123.

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 2-125.

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 2-129.

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.

> **Note:** The `add` and `remove` methods cannot be used for adding and removing attribute properties (specified using the `<search:properties>` element). Only the `overwrite` method can be used for updating the attribute properties.

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 singleSignOnSetting

Updates a Single Sign-On setting.

Syntax

```
update singleSignOnSetting --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update singleSignOnSetting -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 "singleSignOnSetting" on page 2-132.

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 `OSSO`:

```
SES>update singleSignOnSetting --NAME=OSSO --INPUT_FILE=sso_update.xml --UPDATE_
METHOD=overwrite

The object "[name=OSSO]" 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 2-11

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 2-135.

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 3-31.

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 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 singleSignOnSetting

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 2-137.

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 2-165.

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 2-167.

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 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 2-171.

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 suggContent

Sets suggested content configuration parameters from an XML description.

Syntax

```
update suggContent --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update suggContent -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 "suggContent" on page 2-173.

Example

This example updates the suggested content settings:

```
SES>update suggContent --INPUT_FILE=suggContent.xml

The object "suggContent" was successfully updated.
```

update suggContentProvider

Changes the properties of a suggested content provider from an XML file.

See Also

updateAll suggContentProvider

Syntax

```
update suggContentProvider --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update suggContentProvider -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 "suggContentProvider" on page 2-174.

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 suggested content provider *Provider1*:

```
SES>update suggContentProvider --NAME=Provider1 --UPDATE_METHOD=overwrite --INPUT_
FILE=suggcontentproviders.xml

The object "[name=Provider1]" was successfully updated.
```

update suggestion

Updates a suggestion.

See Also

updateAll suggestion

Syntax

```
update suggestion --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_
METHOD=method
```

or

```
update suggestion -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 "suggestion" on page 2-179.

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 `suggestion_1`:

```
SES>update suggestion --NAME=suggestion_1 --INPUT_FILE=suggestion_update.xml
--UPDATE_METHOD=overwrite

The object "[name=suggestion_1]" 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 2-181.

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 tag

Changes the properties of a tag from an XML file.

See Also

updateAll tag

Syntax

```
update tag --NAME=object_name --INPUT_FILE=xml_filename --UPDATE_METHOD=method
```

or

```
update tag -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 "tag" on page 2-185.

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 OSES tag:

```
SES>update tag --NAME="OSES" --UPDATE_METHOD=overwrite --INPUT_FILE=tags.xml

The object "[name=OSES]" was successfully updated.
```

update tagging

Updates the tagging configuration.

Syntax

```
update tagging --UPDATE_METHOD=method --INPUT_FILE=xml_filename
```

or

```
update tagging -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 "tagging" on page 2-183.

Example

This example updates the tagging configuration:

```
SES>update tagging --UPDATE_METHOD=add --INPUT_FILE=tagging.xml

The object "tagging" 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 2-187.

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 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 2-20.

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 boostedUrl

Changes the properties of all boosted URLs described in an XML document.

See Also

update boostedUrl

Syntax

```
updateAll boostedUrl --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_
FOUND_METHOD=action]
```

or

```
updateAll boostedUrl -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "boostedUrl" on page 2-27.

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 all the boosted URLs described in boost.xml:

```
SES>updateAll boostedUrl --UPDATE_METHOD=overwrite --NOT_FOUND_METHOD=ignore
--INPUT_FILE=boost.xml

updateAll operation succeeded for type "boostedUrl".

  2 object(s) with status UPDATE_SUCCEEDED
```

updateAll classification

Sets one or more parameters for all the classifications from an XML description.

See Also

update classification

Syntax

```
updateAll classification --INPUT_FILE=xml_filename --UPDATE_METHOD=method  [--NOT_
FOUND_METHOD=action]
```

or

```
updateAll classification -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "classification" on page 2-29.

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 classifications:

```
SES>updateAll classification --UPDATE_METHOD=add --NOT_FOUND_METHOD=create
--INPUT_FILE=classification2.xml

updateAll operation succeeded for type "classification".

  2 object(s) with status UPDATE_SUCCEEDED
```

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 2-36.

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 docServiceInstance

Sets one or more parameters of all document service instances described in an XML document.

See Also

update docServiceInstance

Syntax

```
updateAll docServiceInstance --INPUT_FILE=xml_filename --UPDATE_METHOD=method
[--NOT_FOUND_METHOD=action] [--ENCRYPT_KEY=key]
```

or

```
updateAll docServiceInstance -i xml_filename -a method [-t action] [-e key]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "docServiceInstance" on page 2-45.

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

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 two document service instances:

```
SES>updateAll docServiceInstance --UPDATE_METHOD=overwrite --NOT_FOUND_
METHOD=create --INPUT_FILE=docserviceinstance.xml --ENCRYPT_KEY=key2encrypt

updateAll operation succeeded for type "docServiceInstance".

  2 object(s) with status UPDATE_SUCCEEDED
```

updateAll docServicePipeline

Sets one or more parameters of all document services pipelines described in an XML document.

See Also

update docServicePipeline

Syntax

```
updateAll docServicePipeline --INPUT_FILE=xml_filename --UPDATE_METHOD=method
[--NOT_FOUND_METHOD=action]
```

or

```
updateAll docServicePipeline -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "docServicePipeline" on page 2-51.

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. It also appends new service instances described in the XML to the end of the execution list, so that the new instances are executed last.
- `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 document service pipelines:

```
updateAll docServicePipeline --UPDATE_METHOD=remove --NOT_FOUND_METHOD=create
--INPUT_FILE=docservicepipeline.xml

updateAll operation succeeded for type "docServicePipeline".

  2 object(s) with status UPDATE_SUCCEEDED
```

updateAll facetTree

Updates all the facet trees described in an XML document.

See Also

update facetTree

Syntax

```
updateAll facetTree --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_
METHOD=action]
```

or

```
updateAll facetTree -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "facetTree" on page 2-54.

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. It also appends new service instances described in the XML to the end of the execution list, so that the new instances are executed last.
- `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 facet trees:

```
updateAll facetTree --UPDATE_METHOD=remove --NOT_FOUND_METHOD=create --INPUT_
FILE=facettree.xml

updateAll operation succeeded for type "facetTree".

  2 object(s) with status UPDATE_SUCCEEDED
```

updateAll indexProfile

Sets one or more parameters of all index profiles described in an XML document.

See Also

update indexProfile

Syntax

```
updateAll indexProfile --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_
FOUND_METHOD=action]
```

or

```
updateAll indexProfile -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "indexProfile" on page 2-74.

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. It also appends new service instances described in the XML to the end of the execution list, so that the new instances are executed last.
- `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 index profiles:

```
SES>updateAll indexProfile --UPDATE_METHOD=remove --NOT_FOUND_METHOD=create
--INPUT_FILE=indexprofile.xml

updateAll operation succeeded for type "indexProfile".

  2 object(s) with status UPDATE_SUCCEEDED
```

updateAll lexer

Sets one or more parameters of all lexers described in an XML document.

See Also

update lexer

Syntax

```
updateAll lexer --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_
METHOD=action]
```

or

```
updateAll lexer -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "lexer" on page 2-79.

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. It also appends new service instances described in the XML to the end of the execution list, so that the new instances are executed last.
- `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 three lexers:

```
SES>updateAll lexer --UPDATE_METHOD=remove --NOT_FOUND_METHOD=create --INPUT_
FILE=lexer.xml

updateAll operation succeeded for type "lexer".

  3 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 2-94.

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 2-125.

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 2-129.

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.

> **Note:** The add and remove methods cannot be used for adding and removing attribute properties (specified using the <search:properties> element). Only the overwrite method can be used for updating the attribute properties.

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 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 2-125.

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 singleSignOnSetting

Sets one or more parameters for the Single Sign-On settings from an XML description.

See Also

update singleSignOnSetting

Syntax

```
updateAll singleSignOnSetting --INPUT_FILE=xml_filename --UPDATE_METHOD=method
[--NOT_FOUND_METHOD=action] [--ENCRYPT_KEY=key]
```

or

```
updateAll singleSignOnSetting -i xml_filename -a method [-t action] [-e key]
```

Parameters

xml_filename

Path to the XML document that configures the object. See "singleSignOnSetting" on page 2-132.

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 singleSignOnSetting object described in singleSignOnSetting_update.xml:

```
SES>updateAll singleSignOnSetting --UPDATE_METHOD=add --NOT_FOUND_METHOD=create
--INPUT_FILE=singleSignOnSetting_update.xml --ENCRYPT_KEY=key2encrypt

updateAll operation succeeded for type "singleSignOnSetting".

  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 2-165.

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 2-167.

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 one 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 2-171.

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 storage area:

```
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 suggContentProvider

Sets one or more parameters for all the suggested content providers from an XML description.

See Also

update suggContentProvider

Syntax

```
updateAll suggContentProvider --INPUT_FILE=xml_filename --UPDATE_METHOD=method
[--NOT_FOUND_METHOD=action]
```

or

```
updateAll suggContentProvider -i xml_filename -a method [-e key] [-t action]
```

Parameters

xml_filename

Path to the XML document that configures the object. See "suggContentProvider" on page 2-174.

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 suggested content provider and creates one suggested content provider:

```
SES>updateAll suggContentProvider --UPDATE_METHOD=add --NOT_FOUND_METHOD=create
--INPUT_FILE=suggcontentprovider_update.xml

updateAll operation succeeded for type "suggContentProvider".

  1 object(s) with status UPDATE_SUCCEEDED
  1 object(s) with status NOT_FOUND_CREATED
```

updateAll suggestion

Sets one or more parameters for all the suggestions from an XML description.

See Also

update suggestion

Syntax

```
updateAll suggestion --INPUT_FILE=xml_filename --UPDATE_METHOD=method  [--NOT_
FOUND_METHOD=action]
```

or

```
updateAll suggestion -i xml_filename -a method [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "suggestion" on page 2-179.

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 suggestions:

```
SES>updateAll suggestion --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_
FILE=suggestions2.xml

updateAll operation succeeded for type "suggestion".

  2 object(s) with status UPDATE_SUCCEEDED
```

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 2-181.

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
```

updateAll tag

Sets one or more parameters for all the tags from an XML description.

See Also

update tag

Syntax

```
updateAll tag --INPUT_FILE=xml_filename --UPDATE_METHOD=method [--NOT_FOUND_
METHOD=action]
```

or

```
updateAll tag -i xml_filename -a method [-e key] [-t action]
```

Parameters

xml_filename
Path to the XML document that configures the object. See "tag" on page 2-185.

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 tag and creates another tag:

```
SES>updateAll tag --UPDATE_METHOD=add --NOT_FOUND_METHOD=create --INPUT_FILE=tags_
update.xml

updateAll operation succeeded for type "tag".

  1 object(s) with status UPDATE_SUCCEEDED
  1 object(s) with status NOT_FOUND_CREATED
```

4

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 and schedule.

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
queryUIFacets
resultList
schedule
singleSignOnSetting
skinBundle
suggContentProvider
tagging
```

<objectKey xmlns="">
Describes the object key for a creatable object type:

```
clusterTree
identityPlugin
schedule
singleSignOnSetting
suggContentProvider
```

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
<	&lt;
>	&gt;
"	&quot;

<decryptionKey xmlns="">

Contains a decryption key for `identityPlugin` and `suggContentProvider` 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 1-6.

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>
   <objectXML xmlns="">
   <decryptionKey xmlns="">
   <credentials xmlns="">
      <password>
      <userName>
   <attachments xmlns="">
      <resourceName>
      <resourcePath>
   <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
boostedUrl
classification
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
```

```
suggestion
suggLink
tag
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
`<`	`&lt;`
`>`	`&gt;`
`"`	`&quot;`

<decryptionKey xmlns="">
Contains a decryption key for these object types:

```
identityPlugin
proxyLogin
source
suggContentProvider
```

<credentials xmlns="">
Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See "Providing Credentials" on page 1-6.

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
boostedUrl
classification
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

<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
<	&lt;
>	&gt;
"	&quot;

<decryptionKey xmlns="">

Contains a decryption key for these object types:

```
identityPlugin
proxyLogin
source
suggContentProvider
```

<credentials xmlns="">

Provides the credentials for the Oracle SES administrator. Credentials are required only when the session is stateless. See "Providing Credentials" on page 1-6.

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>searchsys</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
queryUIFacets
resultList
schedule
singleSignOnSetting
skinBundle
suggContentProvider
tagging
```

<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
singleSignOnSetting
suggContentProvider
```

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 1-6.

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>
  <credentials xmlns="">
    <password>
    <userName>
  <locale xmlns="">
```

Element Descriptions

<delete xmlns="http://search.oracle.com/Admin">
Describes deletion of an administrative object. It contains these elements:

```
classification
creatableType
objectKey
credentials
locale
suggestion
```

<creatableType xmlns="">
Contains one of these creatable types:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
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 1-6.

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:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

<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 1-6.

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>
   <credentials xmlns="">
      <password>
      <userName>
   <controls xmlns="">
      <controlName>
      <controlValue>
   <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:

```
altWord
boostedUrl
classification
clusterTree
docServiceManager
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

<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 1-6.

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`, `source`, and `suggContentProvider` 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 1-6.

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>searchsys</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="11.2.2.2.0"
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:

```
altWord
boostedUrl
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
tag
```

<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`, `source`, and `suggContentProvider` 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 1-6.

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>searchsys</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="11.2.2.2.0"
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:

```
altWord
boostedUrl
clusterTree
identityPlugin
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContentProvider
suggestion
suggLink
```

```
tag
```

<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`, `source`, and `suggContentProvider` 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 1-6.

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="11.2.2.2.0"
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 1-6.

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>
   <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 1-6.

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>11.2.2.2.0</return>
</ns2:getAPIVersionResponse>
```

```
<ns2:getAPIVersionResponse xmlns:ns2="http://search.oracle.com/Admin">
   <version>11.2.2.2.0</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>
   <credentials xmlns="">
      <password>
      <userName>
   <locale xmlns="">
```

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
singleSignOnSetting
skinBundle
suggContentProvider
```

<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 1-6.

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="11.2.2.2.0"
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:

```
autoSuggestion
clustering
clusterTree
identityPlugin
index
indexOptimizer
resultList
schedule
singleSignOnSetting
skinBundle
suggContentProvider
tagging
```

<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 1-6.

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="11.2.2.2.0"
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
singleSignOnSetting
skinBundle
suggContentProvider
```

<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 1-6.

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>searchsys</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="11.2.2.2.0"
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>
   <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>searchsys</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:

```
autoSuggestion
indexOptimizer
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 1-6.

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 1-6.

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>
   <objectXML xmlns="">
   <decryptionKey xmlns="">
   <credentials xmlns="">
      <password>
      <userName>
   <attachments xmlns="">
      <resourceName>
      <resourcePath>
   <controls xmlns="">
      <controlName>
      <controlValue>
   <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:

```
altWord
autoSuggestion
boostedUrl
clustering
clusterTree
crawlerSettings
facetTree
index
indexOptimizer
indexProfile
languageBasedTokenization
```

```
lexer
partitionConfig
proxy
proxyLogin
queryConfig
resultList
schedule
searchAttr
singleSignOnSetting
skinBundle
source
sourceGroup
sourceType
storageArea
suggContent
suggContentProvider
suggestion
suggLink
tag
tagging
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
<	&lt;
>	&gt;
"	&quot;

<decryptionKey xmlns="">

Contains a decryption key for `proxyLogin`, `source`, and `suggContentProvider` 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 1-6.

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.

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:

```
altWord
boostedUrl
clusterTree
facetTree
indexProfile
lexer
proxyLogin
schedule
searchAttr
singleSignOnSetting
source
sourceGroup
sourceType
suggContentProvider
suggestion
suggLink
tag
```

<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
<	&lt;
>	&gt;
"	&quot;

<decryptionKey xmlns="">

Contains a decryption key for `proxyLogin`, `source`, and `suggContentProvider` 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 1-6.

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.

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>
```

A

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 1-6
- *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=\"11.2.2.2.0\"
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=\"11.2.2.2.0\"
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=\"11.2.2.2.0\"
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`.

```
#!/bin/sh

CLASSPATH=.:ses_home/search/lib/search_adminapi_wsclient.jar:java_
home/jre/lib/rt.jar

# Compile
java_home/bin/javac -cp $CLASSPATH CreateWebSource.java

# Run
java_home/jre/bin/java -cp $CLASSPATH CreateWebSource $@
```

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 name, which is, `searchsys`.
- **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 searchsys
password http://example.com/index.htm
```

B

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 1-13

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.

Index

A

about operator, 2-182
ACL policies, 2-153
activate command, 3-1
activate service request, 4-3
address (Web service), 1-6
administration API run-time errors, B-12
administration object errors, B-10
administrator log-in, 1-4
AL32UTF8 character set, 2-84
alternate words, *See* altWord
alternate_spelling attribute, 2-83
altWord
  creating, 3-17, 3-41, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-20
  returning object keys, 3-88, 4-35
  updating, 3-139, 3-181, 4-54, 4-58
and operator, 2-182
API run-time errors, B-12
API version, 3-90, 4-37
applications, searching, 2-137
ASP.NET, 2-xiii, 1-5
asset files, 2-12, 2-135
attribute ranking, 2-115
attributes
  alternate_spelling, 2-83
  base_letter, 2-83
  base_letter_type, 2-83
  composite, 2-83
  continuation, 2-81
  custom search, 2-129, 2-132
  endjoins, 2-82
  in search results, 2-24, 2-123, 2-179, 2-183, 2-185
  new_german_spelling, 2-83
  newline, 2-83
  numgroup, 2-81
  numjoin, 2-81
  printjoins, 2-81
  punctuations, 2-81
  skipjoins, 2-82
  startjoins, 2-82
  whitespace, 2-82
authentication
  plug-ins, 2-65
  self-service, 2-153
  time-out, 2-100
authorization
  plug-ins, 2-147
  time-out, 2-100
authorizedPrincipal
  creating, 3-43
  object description, 2-22
autoSuggestion
  exporting, 3-82, 4-25
  returning current state, 4-42
  starting, 3-134, 4-50
  updating, 4-54

B

base_letter attribute, 2-83
base_letter_type attribute, 2-83
BASIC_LEXER, 2-81
boostedUrl
  creating, 3-18, 3-42, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-27
  returning object keys, 4-35
  updating, 3-141, 3-182, 4-54, 4-58
boundary rules, 2-60, 2-99, 2-106, 2-154
BT (broader term), 2-188
Bubbling library, 2-12
bye, *See* quit command

C

cache files, 2-172
cascading style sheets, 2-12, 2-124, 2-135
character sets, 2-6
  Japanese, 2-84
characters
  continuation, 2-81
  numgroup, 2-81
  numjoin, 2-81
  printjoin, 2-81
  punctuation, 2-81
  skipjoin, 2-82
  specifying for newline, 2-83

specifying for whitespace, 2-82
startjoin and endjoin, 2-82
CHINESE_LEXER, 2-83
CHINESE_VGRAM_LEXER, 2-83
classification
creating, 3-44, 4-6, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86
returning object keys, 3-88, 4-35
updating, 3-142, 3-183
classificationMappings
exporting, 3-82
updating, 3-143
cluster nodes, *See* clustering
cluster trees, *See* clusterTree
clustering
activating, 3-5, 4-3
deactivating, 3-64, 4-13
exporting, 3-82, 4-25
object description, 2-29, 2-31, 2-33
pipeline, 2-42
returning current state, 3-103, 3-104, 4-42
updating, 3-140, 3-144, 4-54
clusterTree
activating, 3-6, 4-3
creating, 3-19, 3-20, 3-45, 4-6, 4-9
deactivating, 3-65, 4-13
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
object description, 2-36
returning current state, 3-91, 3-105, 3-124, 4-39, 4-42, 4-45
returning object keys, 3-88, 4-35
updating, 3-145, 3-184, 4-54, 4-58
collaboration repositories, searching, 2-137
composite attribute
BASIC_LEXER, 2-83
composite word index
creating for German or Dutch text, 2-83
configuration settings
query results, 2-96, 2-102
Config.xsd, 1-11
CONNECTION option, 1-2, 1-4
CONNECTION_LIST option, 1-4
content management repositories, searching, 2-137
continuation attribute, 2-81
cookies, 2-158
count method, 2-98
crawler
character sets, 2-6
documents formats, 2-2
languages, 2-5
log configuration, 2-42
log files, 2-172
scheduling, 2-125
crawler configuration, 2-143, 2-148, 2-155
See also crawlerSettings
crawler settings
See crawlerSettings
crawlerSettings
exporting, 3-82, 4-25
object description, 2-40
updating, 3-146, 4-54
creatable types
listed, 1-9
retrieving object keys, 3-88
create command, 3-1
create service request, 4-6
createAll command, 3-1
createAll service request, 4-9
credentials, 4-48
custom attributes, 2-129, 2-132
customizations, user interface, 2-135

D

dash, using in commands, 1-5
data sources, 2-137
database storage, 2-171
database tables, searching, 2-137
deactivate command, 3-2
deactivate service request, 4-13
delete command, 3-2, 3-75
delete service request, 4-15
deleteAll command, 3-2, 3-76
deleteAll service request, 4-18
deleteList command, 3-2, 3-78
deleteList service request, 4-21
dependent object errors, B-13
dictionary, 2-98, 2-104
directives, 2-12
disk quotas
storageArea, 2-171
display URLs, 2-97, 2-103
do you mean message, 2-21
docServiceInstance
creating, 3-21, 3-46
object description, 2-45
updating, 3-147, 3-185
docServiceManager
creating, 3-22, 3-47
object description, 2-48
returning object keys, 3-88
docServicePipeline
creating, 3-23, 3-48
object description, 2-51
updating, 3-148, 3-186, 3-187
document formats, 2-2
document ranking, 2-115
document service instance, 2-45
document service manager, 2-48
document service pipeline, 2-42, 2-51
document types, 2-63
documents, searching, 2-137
Dutch
composite word indexing, 2-83

E

e-mail display URL, 2-97, 2-103

encryption, 2-9
endjoins attribute, 2-82
endpoint, 1-5
English dictionary, 2-98, 2-104
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, 3-80, 3-82
export service request, 4-25
exportAll command, 3-84
exportAll service request, 4-28
exporting object descriptions, 3-87
exportList command, 3-86
exportList service request, 4-31

F

faceted search, 2-54
facetTree
  object description, 2-54
faults
  See errors
federated search, trusted entities, 2-94
federated sources, 2-99, 2-138, 2-140
federation endpoints, 2-94
file display URL, 2-97, 2-103, 2-142
file size, 2-171
file sources, 2-141
files, searching, 2-137
filter rules, 2-139
forms, 2-160
fragmentation, index, 2-69, 2-71
Freemarker templates, 2-11

G

German
  alternate spelling attribute, 2-83
  composite word indexing, 2-83
  new spelling, querying with, 2-83
getAllObjectKeys command, 3-88
getAllObjectKeys service request, 4-35
getAllStates command, 3-2
getAllStates service request, 4-39
getAPIVersion command, 3-90
getAPIVersion service request, 4-37
getState command, 3-2
getState service request, 4-42
getStateList command, 3-2
getStateList service request, 4-45
globalBoundaryRules
  object description, 2-60
  updating, 3-150
globalDocumentTypes
  object description, 2-63
  updating, 3-151
globalization, 2-4
graphics, 2-12

H

hash partitioning, 2-87, 2-108, 2-110
help command, 1-3
hit count method, 2-98
HTML forms, 2-160
hyphen, using in commands, 1-5

I

identity plug-ins, *See* identityPlugin
identityPlugin
  activating, 3-7, 4-3
  creating, 3-25, 3-50, 4-6, 4-9
  deactivating, 3-66, 4-13
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-65
  returning current state, 3-93, 3-107, 3-126, 4-39, 4-42, 4-45
  returning object keys, 3-88, 4-35
images, 2-12
implementation class, 1-6
inc_footer.ftl, 2-16, 2-18
inc_header.ftl, 2-16
inc_logo_querybox.ftl, 2-16, 2-17
index
  exporting, 3-82, 4-25
  object description, 2-69
  returning current state, 3-109, 4-42
  updating, 3-152, 4-54
index fragmentation, 2-69
index optimizer, *See* indexOptimizer
index refresh, 2-125
indexOptimizer
  activating, 3-8, 4-3
  deactivating, 3-67, 4-13
  exporting, 3-82, 4-25
  object description, 2-71
  returning current state, 3-110, 4-42
  starting, 3-135, 3-137, 4-50
  updating, 3-153, 4-54
indexProfile
  creating, 3-26, 3-51, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-74
  returning object keys, 3-88, 4-35
  updating, 3-154, 3-188, 4-54, 4-58
interactive sessions, 1-2
invalid input errors, B-1
invalid state errors, B-11

J

JA16EUC character set, 2-84
JA16EUCTILDE character set, 2-84

JA16EUCYEN character set, 2-84
JA16SJIS character set, 2-84
JA16SJISTILDE character set, 2-84
JA16SJISYEN character set, 2-84
Japanese character sets supported, 2-84
JAPANESE_LEXER, 2-84
JAPANESE_VGRAM_LEXER, 2-84
Java example, A-1
Javascript libraries, 2-12

K

key names, retrieving, 3-88, 4-35
KOREAN_MORPH_LEXER, 2-85

L

language codes
  crawler, 2-5
  product, 2-4
  query, 2-8
language detection, 2-41
language support, 2-4, 2-7, 2-77, 2-79
language, default, 2-41
languageBasedTokenization
  exporting, 3-82, 4-25
  object description, 2-77
  updating, 3-155, 4-54
lexer
  creating, 3-27, 3-52, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-79
  returning object keys, 3-88, 4-35
  updating, 3-156, 3-189, 4-55, 4-58
lexers
  BASIC_LEXER, 2-81
  Chinese, 2-83
  Japanese, 2-84
  Korean, 2-85
links, depth limit, 2-41
lists of values (LOVs), 2-129, 2-132
localization, 2-4
location, sources types, 2-167
log-in proxies, *See* proxyLogin
log-in service requests, 4-48
logo, custom, 2-14
logout service request, 4-49
logs
  crawler, 2-42, 2-172
  message, 1-13
  operations, 1-13
  querying statistics, 2-99, 2-105
  search statistics, 2-99, 2-105

M

macros, 2-12
mailing list display URL, 2-97, 2-103
message log, 1-13
metadata cluster tree, 2-37
metadata repository, 2-69
MIME types, 2-2

N

near operator, 2-182
nested link limit, 2-41
new_german_spelling attribute, 2-83
newline attribute, 2-83
NT (narrower term), 2-188
numgroup attribute, 2-81
numjoin attribute, 2-81

O

object errors, B-10, B-13
object keys, retrieving, 3-88
objects
  creatable, 1-9
  universal, 1-8
operations log, 1-13
operations, listed by function, 1-14
operators, in search queries, 2-182
optimization, index, 2-69, 2-71
or operator, 2-182
Oracle SES API version, 3-90, 4-37
ORACLE_BASE, xv
ORACLE_HOME, xv
OracleAS Single Sign-On, 2-161

P

partition configuration
  See partitionConfig
partitionConfig
  exporting, 3-82
  object description, 2-87, 2-108, 2-110
  updating, 3-157, 4-55
partitions, 2-172
password encryption, 2-9
PASSWORD option, 1-4
passwords, 4-48
percent signs, using in commands, 1-5
phrase operator, 2-182
port name, 1-6
predefined directives, 2-12
printjoins attribute, 2-81
processes
  for searching, 2-99, 2-100
proxies, *See* proxyLogin
proxyLogin
  creating, 3-28, 3-53, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-94
  returning object keys, 3-88, 4-35
  updating, 3-159, 3-190, 4-55, 4-58
punctuations attribute, 2-81

Q

query configuration, *See* queryConfig
query expansion
  alternate words, 2-20
  suggested links, 2-181
  thesaurus, 2-187
query interface, style sheets, 2-124
query languages, 2-8
query log, 2-99, 2-105
query routing, 2-139
query statistics, 2-99, 2-105
queryConfig
  exporting, 3-82, 4-25
  object description, 2-96, 2-102
  updating, 3-160, 4-55
queryUIConfig
  exporting, 3-82
  updating, 3-161
queryUIFacets
  activating, 3-9, 3-10, 4-3
  deactivating, 3-68, 4-13
  exporting, 3-82
  returning current state, 3-112, 3-113
  updating, 3-162
queryUISourceGroups
  exporting, 3-82
  updating, 3-163
quit command, 1-2
quotas, disk
  See disk quotas

R

ranking documents, 2-115
relevanceRanking
  exporting, 3-82
  object description, 2-115
  updating, 3-164
relevancy boosting, 2-98, 2-104
result list settings, *See* resultList
resultList
  activating, 3-11, 4-3
  deactivating, 3-69, 4-13
  exporting, 3-82, 4-25
  object description, 2-24, 2-123, 2-179, 2-183, 2-185
  returning current state, 3-114, 4-42
  updating, 3-165, 4-55
results, *See* search results
results.ftl, 2-16
robots exclusion, 2-157
RSS feed, 2-14
run-time errors, B-12

S

schedule
  activating, 3-12, 4-3
  creating, 3-29, 3-54, 4-6, 4-9
  deactivating, 3-70, 4-13
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-125
  returning current state, 3-95, 3-115, 3-128, 4-39, 4-42, 4-45
  returning object keys, 3-88, 4-35
  starting, 3-136, 4-50
  stopping, 3-138, 4-52
  updating, 3-166, 3-191, 3-194, 4-55, 4-58
search attributes, *See* searchAttr
search interface customizations, 2-19
search languages, 2-8
search results, 2-96
  identifying top hits, 2-27
  limiting by cluster tree, 2-36
  limiting by source group, 2-165
search statistics, 2-99, 2-105
search statistics log, 2-99, 2-105
searchadmin
  alphabetical command list, 3-1
  command syntax, 1-13
  individual commands, 1-2
  interactive sessions, 1-2
searchAttr
  creating, 3-30, 3-55, 4-6, 4-9
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-129, 2-132
  returning object keys, 3-88, 4-35
  updating, 3-167, 3-192, 4-55, 4-58
secure search, 2-65
self-service authentication, 2-153
service name, 1-5
SES API version, 3-90, 4-37
seslib.ftl, 2-13
session mode
  command-line interface, 1-2
sessions, opening stateful, 4-48
single job mode
  command-line interface, 1-2
single sign-on, 2-161
singleSignOnSetting
  activating, 3-13, 4-3
  deactivating, 3-71, 4-13
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  returning current state, 3-130, 4-39, 4-42, 4-45
  returning object keys, 3-88, 4-35
  updating, 3-168, 3-195, 4-55, 4-58
skin bundles, *See* skinBundle, 2-135
skin bundles, specifying in URLs, 2-14
skinBundle
  activating, 3-14, 4-3
  creating, 3-31, 4-6
  deactivating, 3-72, 4-13
  deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
  directory structure, 2-13
  exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
  object description, 2-135
  returning current state, 3-97, 3-99, 3-117, 3-119, 4-39, 4-42, 4-45
  returning object keys, 3-88, 4-35

updating, 3-169, 4-55
skipjoins attribute, 2-82
source
creating, 3-32, 3-56, 4-6, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
federated, 2-140
file, 2-141
object description, 2-137
returning object keys, 3-88, 4-35
updating, 3-170, 4-55, 4-58
Web, 2-153
source code, Java, A-1
source groups, *See* sourceGroup
source types
list of user-defined, 2-146
See also sourceType
sourceGroup
creating, 3-33, 3-57, 4-6, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
object description, 2-165
returning object keys, 3-88, 4-35
updating, 3-171, 3-196, 4-55, 4-58
sources
search groups, 2-165
See also source, 2-153
user-defined, 2-146
sourceType
creating, 3-34, 3-58, 4-6, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
object description, 2-167
returning object keys, 3-88, 4-35
updating, 3-172, 3-197, 4-55, 4-58
space management
storageArea, 2-171
spaces, using in commands, 1-5
special characters, using in commands, 1-5
spelling correction, 2-98, 2-104
spelling, new German, 2-83
start command, 3-3
start service request, 4-50
startjoins attribute, 2-82
state errors, B-11
state properties, 1-12, 2-9
state, getting current, 3-131, 4-39, 4-42, 4-45
stateful client (Java), 1-6
stateful mode, 1-6
stateful objects, 2-10
stateful sessions, opening, 4-48
stateless client (Java), 1-7
stateless mode, 1-6
State.xsd, 1-11
statistics collection, 2-99, 2-105
statistics log, for searches, 2-99, 2-105
status codes, 1-15
stem operator, 2-182
stop command, 3-3
stop service request, 4-52
storage areas, *See* storageArea
storageArea
creating, 3-35, 3-59, 4-6, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
object description, 2-171
returning object keys, 3-88, 4-35
updating, 3-173, 3-198, 4-55
style sheets, 2-12, 2-135
suggested links, *See* suggLink
suggestion
creating, 3-37, 3-61, 4-7, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
returning object keys, 4-35
updating, 3-176, 3-200, 4-55, 4-58
suggLink
creating, 3-38, 3-62, 4-7, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-25, 4-28, 4-31
object description, 2-181
returning object keys, 3-88, 4-35
updating, 3-177, 3-201, 4-55, 4-58
support bundles, 2-11
SYN (synonym), 2-188
synonyms, 2-187
system files, 2-172

T

table display URL, 2-97, 2-103
tablespaces, 2-171
tag
creating, 3-39, 3-63, 4-7, 4-9
deleting, 3-75, 3-76, 3-78, 4-15, 4-18, 4-21
exporting, 3-80, 3-84, 3-86, 4-28, 4-32
returning object keys, 3-88
updating, 3-178, 3-202, 4-55, 4-58
tagging
activating, 3-16, 4-3
deactivating, 3-74, 4-13
exporting, 3-82
returning current state, 3-123
updating, 3-179, 4-55
template files, 2-135
template library, 2-13
templates, 2-11
thesaurus
creating, 3-40, 4-7
deleting, 3-75, 4-15
exporting, 3-80, 4-25
object description, 2-187
returning object keys, 3-88, 4-35
updating, 3-180, 4-55
thesaurus operator, 2-182
threads
for searching, 2-99, 2-100
threads, for crawling, 2-41
time-out intervals, 2-100
tokenization, 2-77, 2-79

- Chinese, 2-83
- Japanese, 2-84
- Korean, 2-85
- whitespace-delimited languages, 2-81

top *n* results, 2-36
topic cluster tree, 2-37
topic clustering, *See* clustering
translations, 2-7
trusted entities, *See* proxyLogin

U

underscores, using in commands, 1-5
universal types
- listed, 1-8

update command, 3-3
update service request, 4-54
updateAll command, 3-3
updateAll service request, 4-58
URL recommendations, 2-181
URL rewriter, 2-158
URLs
- prioritizing, 2-27

URLs for display, 2-97, 2-103
URLs, nested link limit, 2-41
user interface customizations, 2-135
user names, 4-48
user-defined directives, 2-12
user-defined sources, 2-146
UTF8 character set, 2-84

V

validation, 2-65

W

Web sites, searching, 2-137
Web sources, 2-153
whitespace attribute, 2-82
wildcard characters, 1-5
within operator, 2-182
WSDL address, 1-6

X

XSLT style sheets, 2-124

Y

YUI library, 2-12