

Oracle® Adaptive Access Manager
Command Line Interface Usage Manual
10g (10.1.4.3.0)

December 2007

ORACLE®

Copyright © 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface.....	4
Documentation	4
Before Using the Command Line Interface.....	6
Directory Structure.....	6
Check List.....	6
Command Line.....	7
Basic Command Line Options	8
General Options:.....	8
Export Options:	8
Import Options	9
Advanced Command Line Options.....	10
No entity names for export:.....	10
Importing Multiple Types of Entities in one Transaction	10
Multiple Modules and Extra Options <i>Common v/s Specific</i>	11
Transaction Handling.....	11
Logging	12
log4j.xml Configuration File	12
Sample Log Messages	13

Preface

The Oracle® Adaptive Access Manager Command Line Interface Usage Manual provides information about using the Command Line Interface.

Documentation

The Oracle Adaptive Access Manager 10g documentation includes the following:

- The Oracle® Adaptive Access Manager API Integration Guide, which provides information on natively integrating the client portion of the Adaptive Risk Manager Online solutions. In an API integration, the client application invokes the Adaptive Risk Manager Online APIs directly and manages the authentication and challenge flows.
- The Oracle® Adaptive Access Manager Database Installation Guide (Oracle), which provides information about installing the Adaptive Access Manager schema into an Oracle database. Access to the Adaptive Access Manager schema is a requirement of the Adaptive Access Manager Application Server, which hosts the Adaptive Strong Authenticator and the Adaptive Risk Manager. Note that the Adaptive Manager Access Manager schema needs to be installed into the Oracle database before proceeding to the installation of the proxy.
- The Oracle® Adaptive Access Manager Database Installation Guide for SQL Server 2005, which provides information about installing the Adaptive Access Manager schema into SQL Server 2005. Access to the Adaptive Access Manager schema is a requirement of the Adaptive Access Manager Application Server, which hosts the Adaptive Strong Authenticator and the Adaptive Risk Manager. Note that the Adaptive Manager Access Manager schema needs to be installed into SQL Server 2005 before proceeding to the installation of the proxy.
- The Oracle® Adaptive Access Manager Proxy Integration Guide, which provides programming information and instructions on the installation of the Adaptive Access Manager proxy, one of the components in the Adaptive Access Manager UIO deployment. The Oracle Adaptive Access Manager's Universal Installation Option (UIO) offers multi-factor authentication to Web applications without requiring any change to the application code. The Oracle® Adaptive Access Manager Proxy and The Oracle® Adaptive Access Manager Proxy Web Publishing Configuration are guides specific to the UIO deployment.
- The Oracle® Adaptive Access Manager Proxy Web Publishing Configuration, which provides information on creating web publishing rules and listeners so that Web applications and the WebUIO can be accessible from the Internet. The Oracle Adaptive Access Manager's Universal Installation Option (UIO) offers multi-factor authentication to Web applications without requiring any change to the application code. The Oracle® Adaptive Access Manager Proxy and The Oracle® Adaptive Access Manager Proxy Web Publishing Configuration are guides specific to the UIO deployment.
- The Oracle® Adaptive Risk Manager Online Installation Guide, which provides information on the installation of the administration user interface of Oracle Adaptive Access Manager. Adaptive Risk Manager Online is the administration user interface of Oracle Adaptive Access Manager, a set of web-based administration tools that provides sophisticated fraud monitoring, analysis, and tracking by user location, device, time of day, type of transaction, as well as a host of other factors, and evaluates these factors against a set of customizable rules.

- The Oracle® Adaptive Access Manager LDAP Configuration Guide, which provides information on how to configure the Oracle Adaptive Access Manager Application Server to allow a user to be authenticated via a user identifier and password. The intended audience of this manual are users of WebLogic and Tomcat who want to use LDAP to set up users instead of the functionality in WebLogic and Tomcat.
- The Oracle® Adaptive Access Manager Import/Export Manual, which provides information importing groups, rule templates, and models from the Adaptive Access Manager schema.
- The Oracle® Adaptive Risk Manager Online Customer Care API Guide, which provides information about the Adaptive Risk Manager Online Customer Care API and provides the XML definition for each of the APIs.
- The Oracle® Adaptive Access Manager Database Tables Archiving and Purging Procedure, which provides information on the purge and archive scripts in the Oracle Adaptive Access Manager Database Tables of Microsoft SQL Server 2005. The procedure to trigger the scripts and information on verification and validation of script results are also provided.
- The Oracle® Adaptive Access Manager SQL Server Maintenance Guide, which provides instructions to set up The Oracle Adaptive Access Manager Maintenance Plan to purge and archive scripts in the Oracle Adaptive Access Manager database tables of Microsoft SQL Server 2005. The manual also discusses in detail how to trigger the scripts and provides information on the verification and validation of script results.
- The Oracle® Adaptive Risk Manager™ Administrator's Guide, which provides step-by-step instructions for creating and managing groups, creating models that contain rules, and customizing and managing rules.
- The Oracle® Adaptive Risk Manager™ Dashboard and Reporting Guide, which provides detailed instructions on how to use the dashboard and reporting functionality within the Oracle® Adaptive Risk Manager Online. The Oracle® Adaptive Risk Manager Online includes a dashboard that provides a high-level overview of users and devices that have generated alerts and the alerts themselves, and it contains a comprehensive collection of reports on users, locations, devices, and security alerts.
- The Oracle® Adaptive Risk Manager™ Customer Care Administration Guide, which provides information on creating new customer cases and administering them.

Before Using the Command Line Interface

Directory Structure

The Oracle Adaptive Access Manager Command Line Interface (CLI) package should contain the following directory structure:

```
bharosa_cli
|-- conf (configuragion - XML and PROPERTIES)
|-- docs (cli docs)
|-- lib (vcrypt jar files)
|-- scripts (script to invoke the cli)
|   -- logs (log4j logging files)
`-- thirdparty (thirdparty jar files)
```

Check List

Before we proceed with using the command line interface, the following must be ensured:

- “bharosa_server.properties” must be present in the conf directory of the package. This file contains the environment-specific settings such as database configuration. A sample file “bharosa_server.properties.sample” should be available in the conf directory. You may make a copy of this and set the values according to your requirements.
- You may want to modify the “log4j.xml” present in the conf directory so that the logs are generated in the appropriate files.

Command Line

Go to the scripts directory of the bharosa_cli package. It contains the shell script which invokes the command line interface:

```
$ sh runImportExport.sh
```

Simply giving the above command gives the following “usage” instructions:

```
cli -- import export cli
..... sh runImportExport.sh
..... | -- action < import | export >
..... | ..... < export >
..... | ..... | -- entitycmd < add | delete >
..... | ..... | -- exportmode < zip | file >
..... | ..... | -- includeelements < true | false >
..... | ..... | -- listelemcmd < add | delete | replace >
..... | ..... | -- outdir < path to dest dir >
..... | ..... < import >
..... | ..... | -- batchmode < true | false >
..... | -- module < rules | groups | models | questions | validations >
..... | ..... < groups >
..... | ..... | -- submodule < all | users | alerts | ... >

Usage: cli options [name], [name]...

where:

name = item names to export OR files to import (optional)

and options are:

-action <s> ..... mode of operation: import OR export (required)
-batchmode <s> ..... commit when inserting batches in groups imports (optional)
-entitycmd <s> ..... should the file be exported for deletion (optional)
-exportmode <s> ..... should the files be exported in an archive or extracted
..... individually (optional)
-includeelements <s>
..... whether export the groups with elements or not (optional)
-listelemcmd <s> ..... list items command in the exported file (optional)
-module <s> ..... module to import / export (required)
-outdir <s> ..... directory where the files are to be exported (optional)
-submodule <s> ..... group type to import or export (optional)

Option tags must be separated from their corresponding values by
whitespace, or by an equal sign. Boolean options (options that require no
associated value) may be specified alone (=true), or as 'tag=value' where value
is 'true' or 'false'.
```

Basic Command Line Options

The various options of the command line are as listed above. More detailed information is given in this section:

General Options:

- **action** < import | export >

The valid values for this option are import and export. This is a required option and can not be missing.

- **module** < groups | models | rules | questions | validations >

This option identifies which module we are trying to process. This is also a required option.

- **submodule** < all | alerts | actions | users | vtusers | devices | ips | ipranges | countries | states | cities | connectionSpeed | connectionType | routingType | carrier | sld | tld | asn >

This option can be used with action=export and module=groups. It enables you export all the lists of a certain type, without knowing the names. This option is not mandatory, and its default value is “all”.

Export Options:

- **exportmode** < zip | file >

When using the command line to export any entity, it creates files in the file system. The exportmode option allows you to specify whether the files that are generated by export are to be created directly on the file system, or to be dumped inside a single zip file. The default mode is “zip”, in conformance with the Web UI.

If a zip file, it is named as “<module>_<timestamp>.zip”. In “file” mode, each file is named as “<entity-name>_<timestamp>.xml”.

- **outdir** < path_to_dest_dir >

This option lets you specify where to generate the files (or a single zip file). Default is current working directory.

- **entitycmd** < add | delete >

This option manipulates the XMLs that are generated with the the command specified. The default option is “add”.

- **listelemcmd** < add | replace | delete >

This option is used when exporting lists. It helps to configure the operation for list elements. Default value is “add”.

- **includeelements** < true | false >

This option allows you to specify whether the list being exported should also enumerate the list items. This option can be useful when exporting a list for deletion. Since the list is being deleted, there is no need to include list items.

Import Options

- **batchmode** < true | false >

This option controls the database commits when list items are being imported in a batch. When the batch reaches its limit, the objects are inserted into the database. If `batchmode=true`, this database update is also committed. By default, `batchmode` is set to `false`.

Note: “batchmode” is not to be used in conjunction with importing other modules. It should be used with Lists only.

These options can be followed by entity names in the case of an export, or file names in the case of import. The files can be in XML or Zip format.

Here are some examples:

- Export Rules Rule1 and Rule 2

```
$ sh runImportExport.sh -action export -module rules Rule1 Rule2
```
- Export specific groups Grp1 and Grp2 without elements for delete

```
$ sh runImportExport.sh -action export  
-module groups -includeelements false -entitycmd delete Grp1 Grp2
```
- Export Groups with list command replace

```
$ sh runImportExport.sh -action export -module groups -listelemcmd replace G1 G2
```
- Import a Groups of users in an XML File

```
$ sh runImportExport.sh -action import -module groups abc.xml
```
- Import Rules from Zip or XML File:

```
$ sh runImportExport.sh -action import -module rules Rule.xml Rules.zip
```
- Export Models to DESTDIR, but don't create a zip file

```
$ sh runImportExport.sh -action export -outdir DESTDIR -exportmode file  
-module models Model1 Model2
```
- Import multiple Models from multiple zip files

```
$ sh runImportExport.sh -action import  
-module models ManyModels.zip OneModel.zip
```
- Import multiple questions from multiple zip files

```
$ sh runImportExport.sh -action import  
-module questions ManyQuestions.zip OneQuestions.zip
```
- Import multiple validations from multiple zip files

```
$ sh runImportExport.sh -action import  
-module validations ManyValidations.zip OneValidations.zip
```

NOTE: User may note that inapplicable options will be silently ignored (eg. “outdir” option used for import) and options with lower precedence will be overridden (eg. `listelemcmd` is irrelevant when `includeelements = false`).

Advanced Command Line Options

No entity names for export:

When doing an export, if no entity names are specified, all the entities of that particular module (and submodule) are exported. Thus specifying names is not necessary for export.

Examples:

- Export all models

```
$ sh runImportExport.sh -action export -module models
```
- Export all user groups

```
$ sh runImportExport.sh -action export -module groups -submodule users
```
- Export all rules:

```
$ sh runImportExport.sh -action export -module rules
```
- Export all questions:

```
$ sh runImportExport.sh -action export -module questions
```
- Export all validations:

```
$ sh runImportExport.sh -action export -module validations
```

Importing Multiple Types of Entities in one Transaction

The above examples cover only those scenarios where the entities to be processed are of the same type. In order to be able to process different types of “modules” together, the command line has been altered to support multiple modules. All entities specified in command are processed in a single transaction, which allows a related set of entities to be used together to ensure “all or nothing” approach.

Examples:

- Import various modules together

```
$ sh runImportExport.sh -action import  
-module groups 5grps.zip  
-module models model1.zip  
-module rules r1.xml
```

Note that the action parameter is not to be repeated, but only the command from the “-module” parameter is repeated as per the different items to be imported. The order of the items supplied in the command line is retained for both, the type of entities, and the files for each entity.

Multiple Modules and Extra Options Common v/s Specific

Support for multiple modules raises many questions:

- What about the extra options ?
- How to specify options common to all modules ?
- How to specify options specific to a certain module, even though it has been defined as a common option ?

The following things can be kept in mind:

- When writing an import export command, the idea to keep in mind is that “-module” is considered as the beginning of a new set of options. Everything that follows “-module” forms one set of options.
- Everything that is specified before the first “-module” option is taken as a set of common options, which are applied to each “-module”.
- If a certain option is specified as a common option, and is also specified as a module specific option, the specific value will take precedence.

Examples:

- Export everything to “all” directory, but models to “models” directory.

```
$ sh runImportExport.sh -action export -outdir all  
-module models -outdir models  
-module groups  
-module rules
```
- Export Groups G1 and G2 for delete items, and G3 and G4 for replace items:

```
$ sh runImportExport.sh -action export  
-module groups -listelemcmd delete G1 G2  
-module groups -listelemcmd replace G3 G4
```

Transaction Handling

Transaction handling is different from imports and exports.

Import operates strictly in one transaction, except when using batch mode for importing lists. If there is any error in importing any entity for any module, the entire process is rolled back. Thus, no database updates will be committed. User may also note that though import strictly follows one transaction, it does not break down if it encounters invalid items in a list (eg. importing a city with incorrect state or country, etc.) A warning message is logged and the import process continues, ignoring such items.

Export operates on a “best effort” basis. If an export for any entity fails, it continues with the next entity. The reason being that export does not do any database updates. It only selects information from the database and dumps it into files.

Logging

Log4J logging is being performed throughout the package to record various information when the CLI is run. Setting it at an info level will be ideal for end-users.

Log4J has been configured with an appender to log only the messages from the Import-Export package. This can also be done by adding the following appenders and filters in the log4j.xml configuration file.

log4j.xml Configuration File

```
<appender name="ImportExportAppender"
class="org.apache.log4j.DailyRollingFileAppender">
  <param name="File" value="logs/bharosa_server_ImportExport.log" />
  <param name="DatePattern" value="'. 'yyyy-MM-dd-HH" />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [app=ImportExport]
[%t] %c - %m\n" />
  </layout>
</appender>

<logger name="com.bharosa.vcrypt.utility.importExport">
  <level value="INFO"/>
  <appender-ref ref="ImportExportAppender" />
</logger>

<logger name="com.bharosa.vcrypt.utility.client">
  <level value="INFO"/>
  <appender-ref ref="ImportExportAppender" />
</logger>
```

Sample Log Messages

A sample of log messages is shown below.

```
2006-12-29 17:02:10,295 INFO [app=ImportExport] [main]
com.bharosa.vcrypt.utility.client.ImportExportCmdLine - Import Export
Arguments are { Action = [export], Module = [groups], Sub Module =
[null], Delete Entity = [false], Zip File Name = [null], Include list
elements = [true, List items command = [add], Do db update = [true],
Export Mode = [zip], Output Directory = [.] }

2006-12-29 17:02:10,631 INFO [app=ImportExport] [main]
com.bharosa.vcrypt.utility.importExport.ImportExportManager -
Successfully exported Test My City Group

2006-12-29 17:02:10,663 INFO [app=ImportExport] [main]
com.bharosa.vcrypt.utility.importExport.ImportExportManager -
Successfully exported &

2006-12-29 17:02:10,686 INFO [app=ImportExport] [main]
```

Most of the messages in Import Export, so long as the operation proceeds normally, will be at the INFO level. Errors will be logged when an import or an export fails.