



ORACLE® HYPERION ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

RELEASE 11.1.1

IMPORT/EXPORT UTILITY USER'S GUIDE

ORACLE®
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Importing and Exporting Native Directory Data

This section contains the following topics:

- “Overview” on page 1
- “Use Scenarios” on page 2
 - “Move Provisioning Data Across Environments” on page 2
 - “Manage Users and Groups in Native Directory” on page 3
 - “Bulk Provision Users and Groups” on page 3
- “Installing the Import/Export Utility” on page 3
- “Before Starting Import/Export Operations” on page 3
- “Sample importexport.properties File” on page 4
- “Sequence of Operations” on page 4
- “Preparing the Property File” on page 5
- “Product Codes” on page 9
- “Considerations for Setting Filters” on page 9
- “Prerequisites for Running Import/Export Utility from a Remote Host” on page 10
- “Running the Utility” on page 10
- “Import File format” on page 11
 - “XML File Format” on page 11
 - “CSV File Format” on page 15

Overview

The Import/Export Utility, a standalone, command-line utility, is primarily a tool to manage provisioning by facilitating the bulk-provisioning of user and groups with Oracle Hyperion Enterprise Performance Management System product roles. It allows Oracle's Hyperion® Shared Services Administrators to use an XML or CSV file as the source file to create Native Directory

users, groups, and provisioning information. Shared Services Administrators can use the Import/Export Utility to export, import, and validate data related to various entities:

- Users
- Groups and their relationships
- Roles and their relationship with other roles
- User and group provisioning data
- Delegated lists
- Internal identities of users and groups defined in Native Directory

The utility can be used to export data from a source Native Directory into an export file, which can then be updated imported into a target Native Directory. This utility cannot be used to import data into external user directories. Oracle recommends that you run the utility on the computer that hosts Shared Services.

You can use the Import/Export Utility to create, update, replace, and delete users, groups, and roles that originate from Native Directory. You can also use it to modify groups and role relationships. The utility also validates the quality of the files used for import operations.

Components of the Import/Export Utility:

- Batch file to invoke the operation
- Properties file to configure the utility
- Sample XML data file
- Sample CSV (comma-separated values) data file

Use Scenarios

- [“Move Provisioning Data Across Environments” on page 2](#)
- [“Manage Users and Groups in Native Directory” on page 3](#)
- [“Bulk Provision Users and Groups” on page 3](#)

Move Provisioning Data Across Environments

Shared Services Administrators can use Import/Export Utility to move users, groups, and provisioning data across environments, for example from a development environment to a production environment.

Note:

You can also use the Oracle Hyperion Enterprise Performance Management System Lifecycle Management Utility to move provisioning data across environments. See *Oracle Hyperion Enterprise Performance Management System Lifecycle Management Guide*.

Moving data across environments involves these steps:

- Exporting the data from the source environment into an XML or CSV file
- Modifying the XML or CSV file, if needed
- Validating the updated XML or CSV file
- Importing the XML or CSV file into the target environment

Manage Users and Groups in Native Directory

Shared Services Administrators can create an XML or CSV file containing user and group data, which can then be imported into a target Native Directory to manage users and groups. Bulk creation of users and groups involves these steps:

- Creating a properly formatted XML or CSV file that defines users and groups. See [“Import File format” on page 11](#).
- Validating the XML or CSV file
- Importing the XML or CSV file into the target environment

Bulk Provision Users and Groups

Shared Services Administrators can bulk-provision users and groups using the Import/Export Utility. Bulk provisioning involves these steps:

- Exporting the data from Native Directory into an XML or CSV file or creating a properly formatted XML or CSV file
- Modifying the XML or CSV file to include information on role assignment to users and groups
- Validating the XML or CSV file
- Importing the XML or CSV file back into the Native Directory to update it

Installing the Import/Export Utility

An archive containing the utility is installed into `HYPERION_HOME/common/utilities/CSSImportExportUtility`. Extract the contents of the archive into a directory to which the user who performs the import/export operation has read, write, and execute permissions. The extraction process creates the `importexport` directory and copies the required files into it. This directory is referred to as `IMPEX_HOME` in this discussion.

Before Starting Import/Export Operations

- Create a back up of the source Native Directory by exporting data to an LDAP Data Interchange File (LDIF).
- Ensure that all user directories configured in Shared Services (including Native Directory) are running.
- Ensure that Shared Services is running.

- If you are running the Import/Export Utility from a server that does not host Shared Services, verify that the prerequisites in [“Prerequisites for Running Import/Export Utility from a Remote Host”](#) on page 10 are met.

Sample importexport.properties File

```
#import export operations
importexport.css=http://my_server:port/interop/framework/getCSSConfigFile
importexport.cmshost=localhost
importexport.cmsport=28080
importexport.username=admin
importexport.password={CSS}MRcYv323uzxGr8rFdvQLcA==
importexport.enable.console.traces=true
importexport.trace.events.file=trace.log
importexport.errors.log.file=errors.log
importexport.locale=en
# importexport.ssl_enabled = true

# export operations
export.fileformat=xml
export.file=C:/exportNew.xml
export.internal.identities=true
export.native.user.passwords=true
export.provisioning.all=true
export.delegated.lists=false
export.user.filter=*@Native Directory
export.group.filter=*@Native Directory
export.role.filter=*
export.producttype=HUB-9.2.0
#export.provisioning.apps=(HUB=Global Roles)

# import operations
import.fileformat=xml
import.file=C:/exportNew.xml
import.operation=update
import.failed.operations.file=c:/failed.xml
import.maxerrors=0
```

Sequence of Operations

- Preparing the property file. See [“Preparing the Property File”](#) on page 5.
- Exporting the data into an export file. See [“Running the Utility”](#) on page 10.
- (Optional): Modifying the data in the export file. See [“XML File Format”](#) on page 11 and [“CSV File Format”](#) on page 15.
- Validating the import file. See [“Running the Utility”](#) on page 10.
- Importing the data. See [“Running the Utility”](#) on page 10.

Preparing the Property File

The `importexport.properties` file is a Java properties file that the Import/Export Utility uses during runtime to identify the system components to use for the operation.

The `importexport.properties` file contains three sections:

- **Import export operations:** The settings in this section are used during import and export operations. These settings identify the Shared Services instance and the user credentials.
- **Import operations:** This section contains the parameters for import operations.
- **Export operations:** This section contains the parameters for export operations.

► To prepare `importexport.properties` file:

- 1 **Make a backup copy of the `importexport.properties` file, in the `IMPEX_HOME/samples` directory; for example, `C:\hyperion\common\utilities\CSSImportExportUtility\importexport\samples`.**

Note:

Oracle recommends that the `importexport.properties` file used for the operation be stored in `IMPEX_HOME`.

- 2 **Using a text editor, open the `importexport.properties` file. See “[Sample importexport.properties File](#)” on page 4.**
- 3 **Update properties. Typically, you should update the properties in `import export operations` and one other section, depending on the operation you want to perform:**
 - Update `import operations` to import data into Native Directory or to validate an import file.
 - Update `export operations` to export data into an `.xml` or `.csv` file.

Table 1 Properties for Import-Export Operations

Property	Description
import export operations	
<code>importexport.css</code>	The HTTP URI where Shared Services configuration information (<code>getCSSConfigFile</code>) is available on the application server; for example, <code>http://my_server:port/interop/framework/getCSSConfigFile</code> . For import operations, point to the <code>getCSSConfigFile</code> of the target Shared Services. For export operation, point to the <code>getCSSConfigFile</code> of the source Shared Services.
<code>importexport.cmshost</code>	The DNS name or IP address of the machine that hosts Shared Services. Example: <code>myserver</code>
<code>importexport.cmsport</code>	The Shared Services port number.

Property	Description
	Example: 28080
<code>importexport.username</code>	User account with which to access Shared Services. This user must be able to perform update operations in Native Directory. Example: admin
<code>importexport.password</code>	Password of the user identified in <code>importexport.username</code> . The utility encrypts this password if you enter a plain text password. Example: password
<code>importexport.enable.console.trace</code> <code>s</code>	Indicates whether trace information should be displayed in the console where the Import/Export Utility is executed. Set this property to <code>true</code> to display trace information in the console. Example: true
<code>importexport.trace.events.file</code>	The name and location of the trace log file. If you do not plan to capture trace information in a file, do not set this value. Example: impExtrace.log
<code>importexport.errors.log.file</code>	The name and location of the error log file that should capture information on failed transactions during the import or export operation. Note: Import/Export Utility does not create the error log if you do not specify a file name. Example: impExerror.log
<code>importexport.locale</code>	Locale (two-letter language code) to use for the operation. Supported locales are <code>en</code> , <code>fr</code> , <code>it</code> , <code>de</code> , <code>es</code> , <code>pt_BR</code> , <code>nl</code> , <code>ja</code> , <code>ko</code> , <code>zh_CN</code> , <code>zh_TW</code> , <code>ru</code> , <code>tr</code> . The utility attempts to retrieve only data in the specified locale. If data in the specified locale is not available, Native Directory data in the default locale of the server where the utility is run is exported or imported. Example: en
<code>importexport.ssl_enabled</code>	Indicates whether the import/export operation uses SSL connection. Set the value of this property to <code>true</code> for SSL connections. Example: true Note: If using SSL connection, make sure that the value of <code>importexport.cmsport</code> indicates the SSL port where Shared Services is available.
export operations	
<code>export.fileformat</code>	The format of the export file. You can export data into XML or CSV files.

Property	Description
	Example: xml
export.file	<p>Location of the file into which the data is to be exported. Import/Export Utility creates the file as part of the export process.</p> <p>Example: C:/hyperion/common/utilities/CSSImportExportUtility/importexport/export.xml</p>
export.internal.identities	<p>Indicates whether to export the internal identities of Native Directory users and groups.</p> <p>Internal identity, a component of user and group DN, is unique to each user and group. Shared Services uses an auto-generated identifier as the internal identity. Oracle Hyperion Enterprise Performance Management System products use DN for provisioning. Provisioning information becomes invalid if internal identity is not available, or if it was changed.</p> <p>If you are migrating users from one system to another, you must export the internal identity of users and groups to preserve provisioning information.</p> <p>Example: true</p>
export.native.user.passwords	<p>Indicates whether to export the encrypted passwords of Native Directory users.</p> <p>Note: You cannot perform the CREATE import operation if passwords are not specified in the source file.</p> <p>Example: true</p>
export.provisioning.all	<p>Indicates whether to export all provisioning data. Set this property to false to export a subset of the provisioning data by using these properties in tandem:</p> <ul style="list-style-type: none"> ● export.projectnames ● export.applicationnames <p>Alternatively, you can select a subset by setting export.provisioning.apps.</p> <p>Note: The values of these properties are ignored if export.provisioning.all is set to true.</p> <p>Example: true</p>
export.delegated.lists	<p>Indicates whether to export delegated lists.</p> <p>Example: true</p>
export.user.filter	<p>Optional: Filter to use to select users for export.</p> <p>See “Considerations for Setting Filters” on page 9.</p> <p>Example: *</p>
export.group.filter	<p>Optional: Filter to use to select groups for export.</p> <p>See “Considerations for Setting Filters” on page 9.</p> <p>Example: *</p>

Property	Description
<code>export.role.filter</code>	<p>Optional: Filter to use to select roles for export.</p> <p>See “Considerations for Setting Filters” on page 9.</p> <p>Example: *</p>
<code>export.producttype</code>	<p>Optional: A comma-separated list of product types for which roles are to be exported (must be specified as <i>product code-product version</i>). See “Product Codes” on page 9.</p> <p>Example: HAVA-11.1</p>
<code>export.provisioning.apps</code>	<p>A list of applications (in (projectname=application name) format) from which provisioning data is to be exported. Applications names are listed in the Oracle's Hyperion® Shared Services Console.</p> <p>Example: (<i>Planning Project=Plannig_Application_Name</i>) (<i>Foundation=Shared Services</i>)</p>
import operations	
<code>import.fileformat</code>	<p>The format of the import file. You can import data from XML or CSV files.</p> <p>Example: xml</p>
<code>import.file</code>	<p>Location of the file to import or validate.</p> <p>You can import data from XML or CSV files, created through an export operation. If you manually create the file, be sure to format it correctly. Use the sample CSV and XML files in <i>IMPEX_HOME/samples</i> as reference.</p> <p>Example: C:/hyperion/common/utilities/CSSImportExportUtility/importexport/import.xml</p>
<code>import.operation</code>	<p>The option for the import operation. Valid options:</p> <ul style="list-style-type: none"> ● <i>create</i>—Users, groups, and roles are created. Group, role, and provisioning relationships are augmented. ● <i>update</i>—Users, groups, and roles are updated. Group, role, and provisioning relationships are replaced. ● <i>create/update</i>—A create operation is attempted on each entity in the file. If the operation fails, an update operation is attempted. ● <i>delete</i>—Deletes users, groups, and roles. Group, role, and provisioning relationships are deleted. <p>Example: create</p>
<code>import.failed.operations.file</code>	<p>The name and location of the file where the Import/Export Utility should record information on failed transactions.</p> <p>Example: impFailedOps.log</p>

Property	Description
<code>import.maxerrors</code>	<p>Optional: The maximum number of allowable errors during the import operation. The import operation aborts after the limit is reached.</p> <p>Example: 100</p>

4 Save and close the file.

Product Codes

Table 2 EPM System Product Codes

Product Code	Product Name
EDS	Analytic High Availability Services
ESB	Essbase Server
ESBAPP	Essbase Application
ESVP	Oracle Hyperion Smart View for Office, Fusion Edition
HAVA	Oracle's Hyperion Reporting and Analysis
HBR	Oracle's Hyperion® Business Rules
HFM	Oracle Hyperion Financial Management, Fusion Edition
HP	Oracle Hyperion Planning, Fusion Edition
HPS	Oracle Hyperion Performance Scorecard, Fusion Edition
HSF	Oracle Hyperion Strategic Finance, Fusion Edition
HTM	Oracle's Hyperion® Translation Manager
HUB	Shared Services

Considerations for Setting Filters

The Import/Export Utility uses the settings specified in `importexport.properties` to identify the components (Shared Services, Native Directory, and other user directories) to use for the import or export operation.

During an export operation, Import/Export Utility exports users, groups, and roles based on the filters set for each. The filters are independent of each other.

If a user directory is not specified in the `export.user.filter` or `export.group.filter` value, the filter is applicable to only the user directory where the filter condition is first encountered; other user directories are ignored. User directories are searched (encountered) in the order specified in Shared Services. Because roles are available only in Native Directory, directory specification is irrelevant to role filters.

Note:

If a filter is not specified, data is not exported. *, which is the default filter, exports all data.

Examples: Setting the value of `export.user.filter`, `export.group.filter`, and `export.role.filter` to `k*@Native Directory` exports all Native Directory users, groups, and roles that have names starting with k.

Setting the value of `export.user.filter`, `export.group.filter`, and `export.role.filter` to * exports all users and groups from the first user directory in the search order (see “Managing User Directory Search Order” in the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*) and all roles from Native Directory.

To export users and groups from a specific user directory, set the value of `export.user.filter` and `export.group.filter` to specify the user directory. For example, to export all users and groups from an LDAP-based user directory called LDAP-West, set the value of these filters to `*@LDAP-West`.

While updating `importexport.properties`, you can specify how you want to access trace information. You can view trace information in the console where the Import/Export Utility is executed or store the information in a trace log file, or choose not to generate trace information. You can also view trace information in the console and record it in a file.

The trace log file can be voluminous. Generate a trace file only if you need to debug the import or export operation. Use the information in the error log to identify failed transactions in the trace file.

Note:

Generating trace information will impact the performance of the Import/Export Utility.

Prerequisites for Running Import/Export Utility from a Remote Host

If the Import/Export Utility is being run from a remote host that does not host Shared Services server:

- Verify that Sun JDK 1.5 is installed on the machine from which the Import/Export Utility is run.
- Update the `JAVA_HOME` declaration in `CSSExport`, `CSSImport`, and `CSSValidate` batch files with the location of Sun JDK 1.5 on the machine from which the Import/Export Utility is run.

Running the Utility

The Import/Export Utility comprises three batch files.

- `CSSExport`

- CSSImport
- CSSValidate

Before running the utility, verify that Oracle's Hyperion® Shared Services is running.

► To run the Import/Export Utility:

- 1 **Open a command prompt window.**
- 2 **Navigate to *IMPEX_HOME*, for example, C:\hyperion\common\utilities\CSSImportExportUtility\importexport.**
- 3 **Execute a command:**
 - To export data, run
`CSSExport.bat importexport.properties`
 - To import data, run
`CSSImport.bat importexport.properties`
 - To validate data, run
`CSSValidate.bat importexport.properties`

Note:

If the `importexport.properties` file is not in the directory from which the command is being executed, be sure to use the appropriate path in the commands.

Summary information about the operations is displayed in the console. If transactions fail, review the error log and trace log to determine the cause of the problem and make necessary corrections.

Import File format

The import source file can be an XML file or a CSV file.

- [“XML File Format” on page 11](#)
- [“CSV File Format” on page 15](#)

XML File Format

The data to be imported or validated using the Import/Export Utility can be formatted using XML elements and attributes.

Sample XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<css_data>
  <user id="Test1" provider="Native Directory">
    <login_name>Test1</login_name>
```

```

    <first_name>Test</first_name>
    <last_name>User1</last_name>
    <description>Test user 1</description>
    <email>jch@example.com</email>
        <internal_id>39e706a46ad531be:-48fd959f:
112005bb52e:-8001</internal_id>
        <password>{SHA}D1E0sCEVJhyNL3ukAwldcwRJCG4=</password>>
    </user>
    <group id="mygroup01" provider="Native Directory">
        <name>mygroup01</name>
        <description>mygroupDescr</description>
        <internal_id>39e706a46ad531be:-48fd959f:
112005bb52e:-8000
        </internal_id>
    </group>
    <group_members group_id="G1">
        <group id="CONNECT" provider="orcl">
            <name>CONNECT</name>
            <user id="myUser" provider="orcl">
                <login_name="myUser" </login_name">
            </user>
        </group_members>
    <role id="Administrator" product_type="HUB-9.0.0">
        <name>Administrator</name>
        <description>Have unrestricted access</description>
    </role>
    <role_members role_id="Administrator" product_type="HUB-9.0.0">
        <role id="Provisioning Manager" product_type="HUB-9.0.0">
            <name>Provisioning Manager</name>
        </role>
    </role_members>
    <provision project_name="HUB" application_name="Global Roles">
        <roles>
            <user id="Test1" provider="Native Directory">
                <login_name>Test1</login_name>
            </user>
            <role id=Administrator" product_type="HUB-9.0.0">
                <name>Administrator</name>
                <description>Complete access</description>
            </role>
        </roles>
    </provision>
    <delegated_list id="test2">
        <name>test2</name>
        <description>List description</description>
        <manager>
            <user id="admin" provider="Native Directory">
                <login_name>admin</login_name>
            </user>
        </manager>
        <user id="admin" provider="Native Directory">
            <login_name>admin</login_name>
        </user>
        <group id="G1" provider="Native Directory">
            <name>G2</name>
        </group>

```

```
<delegated_list>
</css_data>
```

Table 3 XML Schema for Import Files

Element	Attribute	Description and Example
css_data		Root element of the file (a container for all other elements)
user		A container for attributes of a user
	id	A unique user ID on the user directory (typically, the same as login_name) Example: pturner
	provider	Name of the source user directory Example: Native Directory
	login_name	User's login name Example: pturner
	first_name	User's first name Example: Paul
	last_name	User's last name Example: Turner
	description	User description Example: Administrative User
	email	User's e-mail Example: pturner@example.com
	internal_id	The autogenerated internal identity of the Native Directory user Example: 911
	password	Encrypted password of the user Example: {SHA}W6ph5Mm5Pz8GgiULbPgZG37mj9g=
group_members		A container for the definitions of groups that contain subgroups or users
	group_id	Name of the nested group Example: test-group
group		A container for group attributes
	id	Group identifier. Same as group name Example: testgroup
	provider	Source user directory for the group Example: LDAP-West

Element	Attribute	Description and Example
	name	Group name Example: testgroup
	description	Group description Example: Test group
	internal_id	The autogenerated internal identity of the Native Directory group Example: 611
role		A container for the attributes of a role
	id	Unique role identifier Example: Basic User
	product_type	Product type to which the role belongs (specified as <i>product code-product version</i>) Example: HAVA-11.1.1
	name	Unique role name Example: Basic User
	description	Role description Example: Launch and view business rules and objects
role_members		A container for attributes of aggregated roles
	id	Unique role identifier Example: Basic User
	product_type	Product type to which the role belongs (specified as <i>product code-product version</i>) Example: HAVA-11.1.1
	name	Unique role name Example: Basic User
provision		A container for provisioning information for an application group-application combination This element contains a definition for each user and/or group who is provisioned to a role in a specific application that belongs to an application group
	project_name	The application group to which the application belongs Example: Business Rules
	application name	The application to which the role belongs Example: local host

Element	Attribute	Description and Example
Delegated List		Container for delegated lists. The users and groups that are managed through a list must also be defined within this container
	id	Unique list identifier, typically the same as the delegated list name Example: Basic User
	name	Name of the delegated list Example: MyList1
	description	List description Example: Delegated list for application creators
	manager	Users and groups who manage the list. Each manager definition may contain user and group definitions. The <code>provider</code> identified must be the user directory that contains the manager's account

CSV File Format

The CSV file format is a tabular data format that contains fields separated by commas and enclosed in double quotation marks. The Import/Export utility supports only Excel-compliant CSV files. The CSV files that Excel outputs differ from the standard CSV files:

- Leading and trailing white space is significant.
- Back slashes are not special characters and do not escape anything.
- Quotes inside quoted strings are escaped with double quotes rather than back slashes.

Excel converts data before putting it in CSV format.

Conversions that Excel performs on CSV files:

- Tabs are converted to single spaces.
- New lines are always represented as the UNIX new line ("\n").
- Numbers of greater than 12 digits are represented in truncated scientific notation form.

The Import/Export Utility categorizes the CSV file into the following entities:

- User
- Group
- Role
- Group_children
- Role_children
- Provisioning
- Delegated list

Each section is identified by two mandatory lines: entity and header. The entity line is identified by a predefined entity name preceded by the # character. The header line follows the entity line. The header line is a comma-separated list of predefined attributes for the entity.

The order of attributes in the header line is insignificant. However, the data lines, which follow the header line, must present data in the order in which the header line presents attributes. If data is not to be specified, you use a comma to indicate that a value is not to be set. The entity line, header line, and data lines provide the information required for processing.

Boundaries applied to create, update, and delete operations on CSV files:

- Users, groups, and roles are processed one data line at a time.
- Group members are processed with multiple data lines under one header and one parent group.
- Role members are processed with multiple data lines under one header and one parent role.
- User provisioning is processed with multiple data lines under one header and one group or user.

Error handling is based on the process boundaries. One error is counted for each failure in a process boundary.

Sample CSV file:

```
#user
id,provider,login_name,first_name,last_name,description,email,internal_id,password
admin,Native Directory,admin,admin,none,Administrative User,,911,{SHA}**=
MyDemoTest,Native Directory,MyDemoTest,admin,none,Administrative
User,-,MyDemoTest222,{SHA}**
#group
id,provider,name,description,internal_id
G1,Native Directory,G1,,39e71be:-4859f:11252e:-8000
WORLD,Native Directory,WORLD,All users are members of this group,611
#group_children
id,group_id,group_provider,user_id,user_provider
G1,CONNECT,orcl,,
G1,,myUser,orcl
#group_children
id,group_id,group_provider,user_id,user_provider
G2,G1,Native Directory,,
#group_children
id,group_id,group_provider,user_id,user_provider
G2Test,,,,
#group_children
id,group_id,group_provider,user_id,user_provider
G3,G2,Native Directory,,
#role
id,product_type,name,description
Administrator,HUB-9.0.0,Administrator,Administrators have unrestricted
access
#role_children
id,product_type,role_id,member_product_type
Administrator,HUB-9.0.0,Provisioning Manager,HUB-9.0.0
#provisioning
project_name,application_name,role_id,product_type,user_id,user_provider,gr
```



```

oup_id,group_provider
HUB,Global Roles,Administrator,HUB-9.0.0,TestUser1,Native Directory,,
#delegated_list
id,name,description,manager_id,manager_provider,user_id,user_provider,group
_id,group_provider
test2,test2,testDescription,admin,Native Directory,admin,Native Directory,,
test2,test2,testDescription,admin,Native Directory,,,G2,Native Directory

```

Tables containing attribute descriptions:

- [Table 4](#)
- [Table 5](#)
- [Table 6](#)
- [Table 7](#)
- [Table 8](#)
- [Table 9](#)
- [Table 10](#)

The following user delineation in an import CSV file can be used to create the user `Test_1` in a Native Directory with the login name `Test_1`, first name `New1`, last name `User1`, description `Test User`, e-mail id `Test1@example.com`, internal id `39e706a46ad531be:-48fd959f:112005bb52e:-8001`, and encrypted password `mypwd`:

```

id,provider,login_name,first_name,last_name,description,email,internal_id,p
assword
Test_1,,Test_1,New1,User1,Test User,Test1@example.com,
39e706a46ad531be:-48fd959f:112005bb52e:-8001,mypwd

```

Note:

The utility encrypts plain text passwords specified in the import file.

Table 4 User Entity Attributes

Attribute	Description and Example
id	User's ID Example: admin
provider	(Optional.) Name of the source user directory Example: Native Directory
login_name	User's login name Example: admin
first_name	Optional: User's first name Example: admin
last_name	Optional: User's last name Example: none

Attribute	Description and Example
description	Optional: User description Example: Administrative User
email	Optional: User's e-mail address Example: admin@example.com
internal_id	The autogenerated internal identity of the Native Directory user Example: 911
password	User's password Example: password

The following group delineation in an import CSV file can be used to create the `WORLD` group in a Native Directory with the group ID `WORLD`, description `Contains all users`, and internal ID `611`:

```
id,provider,name,description, internal_id
WORLD,,WORLD,Contains all users,611,
```

Table 5 Group Entity Attributes

Attribute	Description and Example
id	Group identifier Example: testgroup
provider	Source user directory for the group Example: LDAP-West
name	Group name Example: testgroup
description	Optional: Group description Example: Test group
internal_id	The autogenerated internal identity of the Native Directory group Example: 911

The following role delineation in an import CSV file can be used to create an aggregated role in Native Directory with role id `Designer_rep` for product `hava-11.1.1` (Reporting and Analysis, version 11.1.1), role name `Designer_rep`, and description `Report Designer`. Product type indicates the product to which the aggregated role belongs.

```
id,product_type,name,description
Designer_rep,hava-11.1.1,Designer_rep,Report Designer
```

Table 6 Role Entity Attributes

Attribute	Description and Example
id	Role identifier Example: Basic User
product_type	Product type (specified as <i>product code-product version</i>) to which the role belongs Example: HBR-4.1.1.1
name	Role name Example: Basic User
description	(Optional) Role description Example: Launch and view Business rules and objects.

The following child group delineation in an import CSV file can be used to create the nested group `childGp1` with group id `childGp1`. User member of this group is `Test1`. Both the user and group are defined in `Native Directory`:

```
id,group_id,group_provider,user_id,user_provider
childGp1,childGp1,Native Directory,Test1,Native Directory
```

Table 7 Group_Children Entity Attributes

Attribute	Explanation
id	Identifier of the nested group Example: test-group
group_id	Name of the nested group Example: test-group
group_provider	The source user directory of the group. Example: Native Directory
user_id	Unique identifier of a user who belongs to this group Example: pturner
user_provider	The source user directory of the user assigned to the group Example: LDAP-West

The following child role delineation in an import CSV file can be used to create the nested role `Designer_rep`, which belongs to the product `hava-11.1.1` (Oracle's Hyperion Reporting and Analysis, version 11.1.1), and is assigned to the user `Test1`:

```
id,product_type,role_id,member_product_type
Test1,hava-11.1.1,Designer_rep,hub-11.1.1
```

Table 8 Role_Children Entity Attributes

Attribute	Explanation and Example
id	Unique identifier of a user to whom the role is assigned Example: Test1
product_type	Product type (specified as <i>product code-product version</i>) to which the role belongs Example: hava-11.1.1
role_id	Unique role identifier Example: Designer_rep
member_product_type	The product type (specified as <i>product code-product version</i>) to which the child role belongs Example: hava-11.1.1

The following provisioning delineation in an import CSV file can be used to create a role assignment for application name `Global Roles` that is assigned to the application group `test_proj`. The role ID is `Administrator`, which belongs to product type `HUB-11.1.1`. User `Test1` and group `Group1` defined in Native Directory are provisioned with this role.

```
project_name,application_name,role_id,product_type,user_id,user_provider,group_id,group_provider
HUB,Global Roles,Administrator,HUB-11.1.1,Test1,Native Directory,Group1,Native Directory
```

Table 9 Provisioning Entity Attributes

Attribute	Description and Example
app_id	The application to which the role belongs Example: WebAnalysis
product_type	Product type (specified as <i>product code-product version</i>) to which the role belongs Example: hava-11.1.1
role_id	Unique role identifier Example: Provisioning Manager
user_id	Unique identifier of a user who is provisioned to the role Example: pturner
group_id	Unique identifier of a group that is provisioned to the role Example: testgroup

The following delegated list definition in an import CSV file can be used to create delegated list with list id and name `testlist`, and description `my_list`. Users `admin` and `Test1` defined in Native Directory are delegated administrators of this list which allows them to manage group `testGroup` defined on Native Directory.

```

id,name,description,manager_id,manager_provider,user_id,user_provider,group
_id,group_provider
testlist,testlist,my_list,admin,Native Directory,,testGroup,NativeDirectory
testlist,testlist,my_list,Test1,Native Directory,,testGroup,NativeDirectory

```

Table 10 Delegated List Entity Attributes

Attribute	Description and Example
id	The list identifier, typically, the same as the list name Example: testlist
name	Delegated list name Example: testlist
description	Delegated list description Example: my_list
manager_id	Unique identifier of a user or group who manages the list. Each manager must be identified in a separate definition. Example: admin
manager_provider	The user directory that stores the manager's account Example: Native Directory
user_id	Unique identifier of a user member of the list. Each member must be identified in a separate definition. Example: pturner
manager_provider	The user directory that stores the user member's account Example: Native Directory
group_id	Unique identifier of a group that is a member of the list. Each member must be identified in a separate definition Example: myGroup
group_provider	The user directory that stores the group's account Example: Native Directory

COPYRIGHT NOTICE

EPM System Import/Export Utility User's Guide, 11.1.1

Copyright © 2005, 2008,, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

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.