Oracle® Enterprise Single Sign-on Provisioning Gateway

CLI Guide

Release 11.1.1.5.0

E20982-01

March 2011

ORACLE®

Oracle Enterprise Single Sign-on Provisioning Gateway, CLI Guide, Release 11.1.1.5.0

E20982-02

Table of Contents

Abbreviations and Terminology

Following is a list of commonly-used abbreviations and terminology.

Abbreviation or Terminology	Full Name
Administrative Console	ESSO-LM Administrative Console
Agent	ESSO-LM Agent
FTU	First Time Use Wizard
ESSO-Anywhere	Oracle Enterprise Single Sign-on Anywhere
ESSO-PG	Oracle Enterprise Single Sign-on Provisioning Gateway
ESSO-LM	Oracle Enterprise Single Sign-on
ESSO-PR	Oracle Enterprise Single Sign-on Password Reset
ESSO-UAM	Oracle Enterprise Single Sign-on Universal Authentication Manager

About ESSO-PG CLI

The Oracle Enterprise Single Sign-on Provisioning Gateway (ESSO-PG) server exposes a Web service interface that allows any provisioning server to submit instructions to the ESSO-PG server. The ESSO-PG command-line interface (CLI) is supplied as an integration component for provisioning solutions.

This document describes:

- The format of CLI syntax, return values, commands, options, and parameters
- Escaping parameters containing spaces and quotes
- Setting up SSL for the Java CLI
- Examples illustrating the proper usage of CLI commands

This document describes the .NET and Java CLIs. The functionality of both CLIs is almost identical. The minor differences are noted throughout the document.

There are two versions of the Java CLI: one for version 1.5 and one for version 1.4. They behave almost identically. The one noteworthy difference is discussed in Installing the ESSO-PG CLI.

This document does not describe the platform-specific implementation of the CLI.

Installing the ESSO-PG CLI

Refer to the *ESSO-PG Installation and Setup Guide* for detailed information on installing the CLI.

By default, the .NET CLI will be installed unless you choose to customize the installation. There are two installation options for the Java CLI, Java CLI 1.5, and Java CLI 1.4. You can choose to install either one or both.

For Java 1.5 and .NET installations, there are no further steps needed to be taken after installation.

For Java CLI 1.4, you must perform the following additional steps:

1. Copy the files in the *%Passlogix Home%\v-GO PM\Client\CLI\java14* endorsed directory to *%JAVA_HOME%\lib* endorsed. There are five files to copy: *sax.jar*, *dom.jar*, *jaxp-api.jar*, *xalan.jar*, and *xercesImpl.jar*.
2. Run *pmcli.bat* to execute the Java CLI. The following exception will be thrown the first time any command is issued. Ignore this exception:

```
Oct 19, 2005 3:01:56 PM org.apache.xml.security.Init
registerHereFunction

INFO: Unable to patch xalan function table.

java.lang.NoSuchFieldException: m_functions

… (call stack) …
```

Command Syntax

The CLI is the command-line tool used to send provisioning requests to the ESSO-PG Web service.

Differences between .NET and Java CLI

The .NET CLI executable is called **pmcli.exe.**

The Java CLI implementation is in a class library called **pmcli.jar.** A batch file, **pmcli.bat**, is provided to execute this library. On Windows, an environment variable, `%PMCLI_ROOT%`, must be set to point to the location where `pmcli.jar` and its supporting libraries reside before executing the batch file. The Java CLI can also be executed manually without the batch file in the following manner:

```
java -cp <classpath> pmcli.Main <args>
```

It might be necessary to edit the `pmcli.bat` file and redefine the %P% value according to the directions given in the `pmcli.bat` file. The %P% value refers to the path where the properties file is stored. The Java CLI can be customized using the properties file. This file must live along a path without any spaces in the name. By default, the Java CLI is installed on Windows under Program Files, which requires that if a properties file is used, the value of %P% must be set to refer to a directory name without any spaces where the file can be placed.

The CLI uses the following syntax:

```
usage: pmcli [-url service] [-agent name] [-u login id]
[-p password] [-t date/time] [-f inputfile]
[-security <sec_opts>] "operation"
```

The CLI accepts switches in the following format or any combination:

`-arg=value`	Value specified after "="
`-arg` ***value***	Value specified as next argument
`-arg:`*value*	Value specified after ":"
`--arg`	Double dash to start an arg
`/arg`	Forward slash to start an arg

In version 7.0, the CLI supports the following new switches:

Switch	Description
`-u, -p`	Equivalent to –security username=<value> password=<value>
`-f`	Executes batch operations from a file, then exits
`-t`	Alias for –exec. Specifies time to execute provisioning operation.

Modes of Operation

There are three supported modes of operation:

- Command-line mode
- Batch mode
- Interactive mode

Command-line mode

In this mode, you specify the provisioning operation by entering it on the command line. The following provisioning operations are supported:

Operation	Definition
ADD_CREDENTIAL	Add new credential
MODIFY_CREDENTIAL	Modify an existing credential
DELETE_CREDENTIAL	Delete an existing credential
DELETE_USER	Delete SSO user and their stored credentials
STATUS	Get status of a pending instruction
CANCEL	Cancel a pending provisioning instruction
EXT_SEARCH	Search for logon and pending requests
SET_SETTINGS	Change the current storage settings
GET_SETTINGS	Retrieve the current storage settings
GET_SCHEMA	Retrieve the available storage schemas
CHECK_SERVER	Check status of server

Each of these operations and their parameters are described in a later section of this document.

If both a batch file and operation are specified on the command line, batch mode takes precedence.

Batch mode

Batch mode allows you to pass a series of provisioning operations to the CLI in a file specified through the `-f` switch.

Interactive mode

If there is no operation specified on the command line and a batch file is indicated, the CLI enters interactive mode. In this mode, provisioning operations are specified in a shell-like environment until you enter `quit` or `exit`.

Interactive mode supports three additional commands not available in the command-line mode or batch mode:

Command	Description
HELP	List all commands available
`Help [operation]`	Show syntax for a specific command
`QUIT, EXIT, Q, E`	Exit from interactive mode or stop executing the batch

Smart Defaults

If the url, agent, username, or password switch is not specified, the CLI uses the following defaults:

Switch	Default
`-url`	http://localhost/v-GO%20PM%20Service/UP.asmx
`-agent`	The current machine name (on Windows %MACHINENAME%).
`-password`	The CLI will prompt for a password.

Differences between .NET and Java CLI

For security reasons, the .NET CLI will obfuscate the password entered by a user (if the user is prompted for a password). For platform-independent reasons, the Java CLI will not obfuscate the password entered by a user.

Operation Execution

When an operation has been executed by the CLI, it outputs the results to the screen. The format output will depend on the operation executed. In general, the result is as follows:

`[RESULT] ID: [GUID]`		
`[RESPONSE]`		
where:		
`[RESULT]`	The result of the provisioning server.	
	`success`	A request has been successfully created and placed in the directory. The agent processes this request and marks it either success or failure.
	`noSuchRequest`	The request ID does not exist. This applies to the status and cancel operations.
	`CouldNotCancel`	The request is in a state that does not allow it to be canceled. This applies to the cancel operation.
`[GUID]`	The unique identifier of the provisioning instruction that was successfully submitted.	
`[RESPONSE]`	Additional results returned by the particular provisioning instruction. This applies to the status, ext_search, get_settings, and get_schema operations. The results are generally in name-value pair format. This attribute format can be viewed as descriptors for the information being returned.	
In the event of an error, the output will be the exception followed by a descriptive message		
`[exception]: [descriptive error message]`		

Usage

The command, `pmcli -?`, will display usage and syntax information.

Status Results

When the ESSO-LM Agent has finished processing a provisioning instruction, the `Result` attribute of the instruction is set to the result of execution. If the agent fails to process an instruction, the attribute is set to `Failed`, and the `Description` is set to the specific error that occurred. The possible error cases are:

- Failure to decrypt the provisioning instruction
- Failure to delete the requested instruction
- Invalid or unknown instruction type
- Failed to find application specified in instruction
- Failed to treat modify instruction as an add instruction

- Failed to add instruction, credential already exists
- Failed to add instruction, required field not included

Provisioning Operations

The following table lists the specific provisioning operations that can be executed and the specific syntax for each operation:

add_credential	Add a new credential for a given user.
delete_credential	Delete an existing credential associated with a given user.
modify_credential	Modify an existing credential associated with a given user.
delete_user	Delete SSO user and their stored credentials.
status	Get status of pending and submitted provisioning instructions.
cancel	Cancel a pending provisioning instruction.
ext_search	Searches for applications, users, and event log entries.
set_settings	Change the current storage settings.
get_settings	Retrieve the current storage settings.
get_schema	Retrieve the available storage schemas.
check_server	Checks the status of the server (no errors on success).

Parameters

The operation parameters define the specific characteristics for the request. The set of expected parameters are listed per operation. Each parameter consists of a name-value pair specified as follows:

sso_ userid	The user's ID as known by ESSO-PG. This is the ID used by the Provisioning Service to locate the user in the ESSO-PG data store.
sso_ application	The name of the application to add a credential to.
sso_ description	The description of the credential. This field is optional.
sso_app_ userid	The application's user ID field for this credential.
sso_ password	The password field for this credential.
sso_other1	The third field for this credential.
sso_other2	The fourth field for this credential.
command_ id	The GUID submitted by a successful provisioning request.

SET_SETTINGS

The following describes the specific settings for the set_settings operation:

name	A comma-delimited list of storage key names.
value	A comma-delimited list of storage values.

EXT_SEARCH

The following table defines the specific settings for the ext_search operation:

catalog	The catalog to search.
userId	The sso_userid of the user to find (ext_search).
logon	A comma-delimited list of application logon names.
returnLogons	Return a list of GUIDs associating stored credential containers to application templates for the selected user.
returnInstructions	Return a list of pending instructions.
uidMatch	Do an exact or substring match on userId.
startDate	The start date of the event log.
endDate	The end date of the event log.
eventType	The type of event to filter the search on.

Syntax

The syntax describes the parameters and format expected for each operation. The following defines each operation and its syntax:

```
ADD_CREDENTIAL sso_userid sso_application [sso_app_userid]
[sso_password] [sso_description] [sso_other1] [sso_other2]

MODIFY_CREDENTIAL sso_userid sso_application sso_app_userid
[sso_description] [sso_password] [sso_other1] [sso_other2]

DELETE_CREDENTIAL sso_userid sso_application
[sso_app_userid] [sso_password] [sso_other1] [sso_other2]

DELETE_USER sso_userid

STATUS sso_userid command_id

CANCEL sso_userid command_id
```

```
EXT_SEARCH CATALOG=Applications [userId]
```

```
EXT_SEARCH CATALOG=Users [userId] [logon="logon1,logon2,..."]
[returnLogons=true|false] [returnInstructions=true|false]
[uidMatch=substring|equal]
```

If uidMatch is not specified, equal is assumed. If returnLogons and returnInstructions are not specified, false is assumed.

```
EXT_SEARCH CATALOG=EventLog [startDate=mm/dd/yyyy]
[endDate=mm/dd/yyyy] [eventType=amducs]
```

The possible values of eventType are:

a	Add Logon
m	Modify Logon
d	Delete Logon
c	Delete User
u	Cancel Request
s	Status Request

These can be used in combination to return matching events.

```
SET_SETTINGS name="key1,key2,..." value="value1,value2,..."
```

Valid keys can be obtained using GET_SCHEMA. The number of keys and values must be identical. Each key in the name list is paired with its matching value on the value list (based on position).

GET_SETTINGS	There are no parameters for this command.
GET_SCHEMA	There are no parameters for this command.
CHECK_SERVER	There are no parameters for this command.

Escaping a Comma

Parameters that take comma-delimited values support the \ (backslash) as an escape character for commas. For example, to enter the value `CN=USERS,DC=DOMAIN,DC=COM` for the UserPath in AD, you would issue the following command:

```
SET_SETTINGS name="Storage\AD\UserPath"
value="CN=USERS\,DC=DOMAIN\,DC=COM"
```

Commas that are not escaped are treated as delimiters between multiple values or keys.

Setting Up Java for SSL

To set up SSL support for the Java CLI, you must modify a properties file to point to the Java Keystore File root:

1. Download a public version (no private key) of the SSL certificate that will be used. This can be retrieved from the server that is hosting IIS. Save this public certificate as an ssl.cer as follows:
   a. From the server with the SSL certificate, open the Microsoft Management Console by selecting **Start** > **Run**, type **MMC** and click **OK**.
   b. Click **File** > **Add/Remove Certificates Snap-in**. On the **Standalone** tab, click **Add**.
   c. Select the **Certificate** snap-in and click **Add**.
   d. Select **Computer Account** and click **Next**.
   e. Select **Local Computer** and click **Finish**.
   f. Under the **Console Root**, expand **Certificates (Local Computer)**.
   g. Expand **Personal** and click **Certificates**.
   h. Right-click the SSL certificate and select **All Tasks** > **Export**.
   i. On the Certificate Export Wizard panel, click **Next**.
   j. On the Export Private Key panel, click **No, do not export the private key**.
   k. Select the file format you want to use (either DER or BASE-64) and click **Next**.
   l. Browse to locate the file you want to export. Click **Next**.
   m. Save as an ssl.cer file.
   n. Click **Finish**, and then click **OK**. This file will be imported into the java keystore on the client (we will create this next).
2. Verify that JDK 1.42+ is installed on the client workstation. There is a binary called `keytool.exe` that you will use to create the keystore.
3. Create a file called `pmcli.jks` with an alias of `pmssl` as follows:
   a. Run: `keytool -import -trustcacerts -file ssl.cer -alias pmssl -keystore pmcli.jks`
   b. Enter a password for the keystore.
   c. When prompted to trust certificate, click **Yes**.
   d. Copy the **pmcli.jks** file to the folder where **pmcli.ja**r is located.
4. Create a `pmcli.properties` file in the folder defined for property files in `pmcli.bat`.
5. Edit `pmcli.properties` by adding the following line:
   `rmi.ssl.trust.keystore.location=pmcli.jks`
   Save the file.
6. Add the full path to the directory where `pmcli.properties` lives (not the full path to the file) to the CLASSPATH.
7. Run `pmcli.bat` and pass an https URL to the `-url` switch.

Enabling SSL will still allow the CLI to communicate with an http service.

Examples

The following examples demonstrate how the CLI is used.

Switches example:

```
pmcli –username=johns

pmcli –username johns

pmcli –username:johns

pmcli –u:johns

pmcli –u=johns

pmcli –u johns

pmcli /u:johns

pmcli --u:johns
```

The above calls are equivalent and apply to all switches.

Smart defaults example:

```
pmcli –p:Password

url defaults to http://localhost/v-go%20pm%20service/up.asmx

agent defaults to machine name

username is current logged in user

pmcli –u:Administrator –p:Password

url defaults to http://localhost/v-go%20pm%20service/up.asmx

agent defaults to machine name

pmcli –url:http://test.com/v-go%20pm%20service/up.asmx -p:mypassword

agent defaults to machine name

username is current logged in user

pmcli

url defaults to http://localhost/v-go%20pm%20service/up.asmx

agent defaults to machine name

username is current logged in user

password is prompted (CLI will prompt you for a password)
```

This example adds a Lotus Notes credential for the SSO user joeuser:

```
pmcli –url "http://example.com/v-GO PM Service/UP.asmx" –agent "PM
Agent" –username=PMAdmin -password=mysecretpassword add_credential
```

```
sso_userid=joeuser sso_application="Lotus Notes"
sso_app_userid=lotususer sso_password=password123 sso_other1=mydomain
```

The first four switches to the CLI indicate:

- the location of the ESSO-PG Web service
- the identifier for this agent
- the credentials to use to authenticate against the Web service
- the operation and its parameters.

In this case, the SSO user to provision is *joeuser* and a credential was added for Lotus Notes with credentials of *lotususer* and *password123* in the *mydomain* domain.

This example deletes all credentials for the SSO user joeuser:

```
pmcli –url "http://example.com/v-GO PM Service/UP.asmx" –agent "PM
Agent" –username=PMAdmin –password=mysecretpassword delete_user sso_
userid=joeuser
```

This example returns a list of users with provisioned logons and instructions on the system:

```
pmcli –url "http://example.com/v-GO PM Service/UP.asmx" –agent "PM
Agent" –username=PMAdmin –password=mysecretpassword ext_search
catalog=users returnLogons=true returnInstructions=true
```

This example demonstrates how to execute operations from a batch file:

```
pmcli –url:"http://example.com/v-GO PM Service/UP.asmx" –agent:"PM
Agent" –u:PMAdmin –p:mysecretpassword -f=c:\operations.txt
```

The file operations.txt contains provisioning operations, one on each line:

```
add_credential sso_userid=joeuser sso_application="Lotus Notes" ...

add_credential sso_userid=janeuser sso_application="Lotus Notes" ...

delete_credential sso_userid=jackuser sso_application="Lotus Notes"
```

This example demonstrates how to run the CLI in interactive mode:

```
pmcli –url:"http://example.pass.com/v-GO PM Service/UP.asmx" –agent:

"PM Agent" –u:PMAdmin –p:mysecretpassword
```

The CLI will enter interactive mode:

```
Passlogix (R) v-GO PM CLI Version 6.0.0

Copyright (C) Passlogix, Inc. 1998-2005. All rights reserved.

URL: http://example.pass.com/v-GO PM Service/UP.asmx

AGENT: PM Agent”

USERNAME: PMAdmin

EXECUTE: 10/17/2005-15:07:04

-------------------------------------
```

```
Type 'e[xit]' or 'q[uit]' to end session.

HELP

HELP [operation]

operation - displays help information on that operation.

> _
```

The user can enter provisioning operations at the prompt similar to the operations in batch mode until a quit or exit is encountered.

This example demonstrates how to specify when to run the provisioning operation:

Specifying the `-t` switch on the command line followed by a time indicates that the provisioning operation should only be executed by the ESSO-LM Agent on or after the specified time. The operation will exist on the directory service and the ESSO-PG Agent will execute it, but the logon will not be available to the SSO user until the time specified. The format of `-t` is:

```
Java: MM/DD/YYYY-HH:MM:SS

.NET: "MM/DD/YYYY HH:MM:SS"
```