Oracle® Fusion Middleware

Enterprise Single Sign-On Provisioning Gateway Administrator's Guide

11g Release 2 (11.1.2.2)

E37696-01

December 2013

ORACLE®

Oracle Fusion Middleware Enterprise Single Sign-On Provisioning Gateway Administrator's Guide, 11g Release 2 (11.1.2.2)

E37696-01

Copyright © 1998, 2013, Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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

Contents

5 The Provisioning Gateway Node

6 Reports and Logs

7 The Provisioning Gateway CLIs

Preface

Oracle Enterprise Single Sign-On Provisioning Gateway Administrator's Guide explains how to use the Provisioning Gateway Administrative Console to remotely add, modify, and delete application credentials directly within each user's Logon Manager credential store, eliminating the need for local credential capture and granting the user instant access to the target application.

Audience

This document is intended for experienced administrators responsible for the planning, implementation, and deployment of Provisioning Gateway. Administrators are expected to understand single sign-on concepts and be familiar with Internet Information Services, Windows Registry settings, and the Oracle Enterprise Single Sign-On Administrative Console.

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 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, see the following documents in the Oracle Enterprise Single Sign-On Suite documentation set:

- *Release Notes*
- *Oracle Enterprise Single Sign-On Suit Installation Guide*
- *Oracle Enterprise Single Sign-On Suite Administrator's Guide*
- *Oracle Enterprise Single Sign-On Suite Secure Deployment Guide*
- *Oracle Enterprise Single Sign-On Suite User's Guide*
- *Deploying Logon Manager with a Directory-Based Repository*

- *Configuring and Diagnosing Logon Manager Application Templates*
- *Oracle Enterprise Single Sign-On Provisioning Gateway Administrator's Guide*

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.

1

Overview of the Provisioning Gateway Administrative Console

The Provisioning Gateway Administrative Console enables administrators to set up, gather, and manage information from the Provisioning Gateway Web service. The following modules can be accessed from the Provisioning Gateway Administrative Console:

- Settings
- Users
- Reports and Logs

1.1 Accessing the Provisioning Gateway Administrative Console

To access the Provisioning Gateway Administrative Console:

- Open a Web browser and enter the following URL:

  `https://`*`yourserverhost`*`/v-go pm console/logon.aspx`

  where

  `yourserverhost` is the name of the server where you installed Provisioning Gateway.

  The Logon Page opens.

1.1.1 Version Information

The About module provides information about which versions of Provisioning Gateway and Microsoft .NET Framework are installed.

- **Product Version.** Indicates which version of Provisioning Gateway is installed.
- **.NET Framework.** Indicates which version of Microsoft .NET Framework is installed.

1.1.2 Logon Page

Enter your logon credentials to access the Provisioning Gateway Web Service and click **Log On**. The username and password should be the same as the directory authentication credentials. For example, for Active Directory or AD LDS (ADAM), the username would be in the format: *domainname\username*.

For Sun or IBM, the username would be in the format: *uid=username*.

Note: The Provisioning Gateway server recognizes only credentials that it has access to. On Active Directory or AD LDS (ADAM), those recognized credentials are domain accounts. For Sun and IBM, the account must exist in the storage. If no storage has been defined, the account is authenticated against the local accounts on the server where the Web service is running.

1.1.3 Security Settings

Provisioning Gateway can be run without changing the default security settings. Security can be increased by changing several of the settings.

You can edit the Provisioning Gateway security settings through the Microsoft .Net Framework ASP.NET Configuration Settings. These settings are then changed in the Provisioning Gateway configuration files:

- `<local directory>\Provisioning Gateway\Service\web.config`
- `<local directory>\Provisioning Gateway\Console\web.config`

1.1.4 Granting Access to the Provisioning Gateway Administrative Console

By default, all users are denied access to the Provisioning Gateway Administrative Console. You can assign users provisioning rights through the Oracle Enterprise Single Sign-On Administrative Console. You can perform the following actions on users:

- Provisioning a logon (adding, modifying, deleting credentials) for a user. Assign these rights in the **Provisioning** tab of a template.
- Deleting an SSO user. Do this on the **Delete SSO User Right** tab of the Provisioning Gateway node.

Configure these settings and publish them to the repository to grant users access to the Provisioning Gateway Administrative Console.

See the *Oracle Enterprise Single Sign-On Suite Administrator's Guide* for more information on using these settings.

1.1.5 Changing the Encryption Algorithm

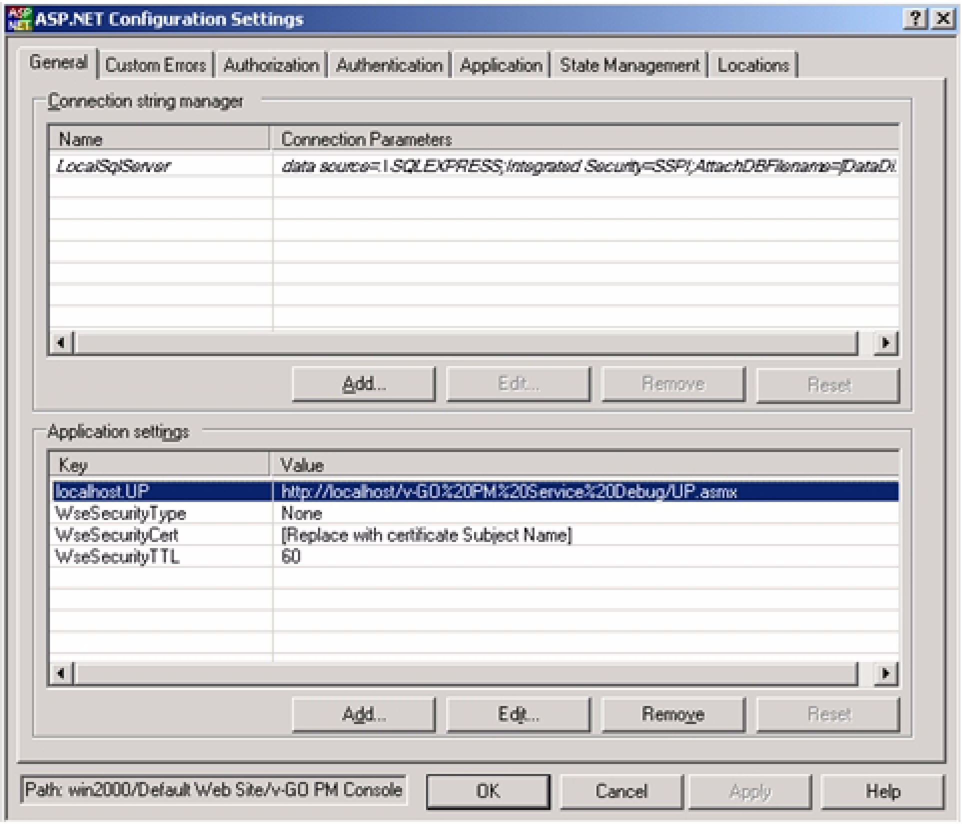
By default, the Provisioning Gateway Web service uses 3DES encryption. To increase security, you can change encryption to AES. In order to enable this feature, you must edit a setting in Oracle Service Properties:

1. Go to **Control Panel** > **Internet Information Services**.
2. Right-click the Provisioning Gateway Service Web site. Select **Properties**.
3. Click the **ASP.NET** tab. Verify that the ASP.NET version is set to 2.0.x. (If it is not set to 2.0, change the setting and click **Apply**.) Click **Edit Configuration**.
4. In the **ASP.NET Configuration Settings** dialog, highlight **EncryptionAlgorithm** and click **Edit**.
5. In the **Value** field, replace 3DES with AES_256. This value causes the Provisioning Gateway Service to use the AES encryption method.

1.1.6 Enabling SSL

For testing purposes, you can enable SSL by changing the `localhost.UP` key in Provisioning Gateway Console Properties:

1. In the **ASP.NET Configuration Settings** dialog, highlight `localhost.UP` and click **Edit**.



Name	Connection Parameters
LocalSqlServer	data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBFilename=|DataDi

Key	Value
localhost.UP	http://localhost/v-GO%20PM%20Service%20Debug/UP.asmx
WseSecurityType	None
WseSecurityCert	[Replace with certificate Subject Name]
WseSecurityTTL	60

Path: win2000/Default Web Site/v-GO PM Console

2. Go to **Control Panel** > **Internet Information Services**. Right-click the Provisioning Gateway Console Web site. Select **Properties**.

3. Click the **ASP.NET** tab. Verify that the ASP.NET version is set to 2.0.x. (If it is not set to 2.0, change the setting and click **Apply**.) Click **Edit Configuration**.

4. In the **Value** field, replace: `http://localhost/Provisioning Gateway Service/UP.asmx` by entering `https://localhost/Provisioning Gateway Service/UP.asmx`.

5. You can now edit the properties for the Provisioning Gateway Service in IIS to turn on SSL.

1.1.7 Setting Permissions

When you install Provisioning Gateway, you must create a specific service account, at the domain level, in order for Provisioning Gateway to function properly. This section describes how to increase security by creating such an account with a specific set of permissions to certain objects within Active Directory.

In order to increase security, Oracle recommends that this service account be created as a member of the Domain Users group. (For the purposes of this document, the service account is named PMSERVICE; however, you can follow any naming convention you choose).

The instructions in this section describe how to:

- Create the service account (PMSERVICE) as a member of the Domain Users group.
- Grant a specific set of permissions to certain objects within Active Directory to the serviced account.
- Create templates for provisioning.
- Provision a user.

Note: The PMSERVICE account must also be a member of the local administrator's group on the IIS server where the Provisioning Gateway server-side components are installed.

You will need an account with Domain Admin and Schema Admin privileges in order to complete certain tasks involving the installation of Logon Manager, extending the schema, installing software, and modifying certain permissions within Active Directory.

1.2 General Recommendations and Notes

Microsoft recommends that you not install Internet Information Server (IIS) on a Domain Controller. Oracle recommends that you install the Provisioning Gateway Server-side components on a member server, not a Domain Controller.

The procedures and recommendations presented in this document have been tested in a controlled environment where the desired results were achieved. Oracle recommends that you test these procedures in a non-production environment that resembles your working network as closely as possible.

The procedures outlined herein involve changes that can affect your entire domain. Specialized policies, trust, inheritance issues, and intra- and inter-site replication issues, particularly as they exist in large enterprises, cannot be fully tested outside of the actual environment.

As with any issues that could affect a large number of users, Oracle recommends a prudent, error-on-the-side-of-caution approach to testing and deploying this product by those who are responsible for installing, configuring, and maintaining it.

2 Provisioning Gateway Settings

This chapter provides information about using the settings that configure Provisioning Gateway.

2.1 Web Service Account Settings

Use the Web Service Account page to set or change the Anonymous Logon for IIS Web Services. The Provisioning Gateway Web service runs as this domain account. The Web Service Account dialog displays the current Anonymous Logon account and provides a logon form for changing this account.

> **Note:** You must be authenticated to the Provisioning Gateway Console as a member of the administrator group of the Provisioning Gateway Web server to change the account.

The Web service account requires the following privileges:

- Read and write access to the Registry path `HKLM\Software\Passlogix`.
- Connect, read, and write access to the storage if Active Directory or AD LDS (ADAM).

To change the Web service account, type in the account **User Name** (in the format `Domain\Username`) and **Password**, confirm the password, and click **Save**.

2.2 Storage Settings

Use the **Storage** page to view or change connection settings for the directory service (Oracle Internet Directory, Microsoft Active Directory, Microsoft AD LDS (ADAM), IBM LDAP Directory, or Sun Directory Server) being used as the repository for Provisioning Gateway data.

When you have completed your changes, click **Save Changes** to apply the new settings to Provisioning Gateway. After the storage settings are saved, you will be prompted to re-authenticate to Provisioning Gateway.

The information on this page is encrypted and saved to the registry under `HKLM\Software\Passlogix\PM\Server\Storage`.

Setting	Value
*Storage Type	Choose one of the following storage locations: ■ Oracle Internet Directory ■ Oracle Directory Server Enterprise Edition ■ Oracle Virtual Directory ■ Sun Directory Server ■ Microsoft Active Directory ■ Microsoft AD LDS (ADAM) ■ IBM LDAP Directory ■ Novell eDirectory
*Server	Enter either the name of the server or the IP address of the server.
*Root DN	The root directory. For example, `DC=mydir,DC=com`.
Provide this setting for Oracle Directories, Active Directory, IBM LDAP Directory, Novell eDirectory, and Sun Directory Server storage only:	
*User Path(s)	The fully-qualified path indicating the location of user accounts. There can be unlimited paths to search. The paths are searched in the order they are entered and are separated by a semicolon (;). For example, `CN=users,DC=mydir,DC=com`
Provide these settings for Active Directory and/or AD LDS (ADAM) storage only:	
Prepend Domain	Select this option to add the user's domain to the username when naming the user's container. For example, for the domain *oracle* and user *jamesk*, the container is named *jamesk* with this flag disabled and *oracle.jamesk* with this flag enabled.
Provide this setting for Active Directory storage only:	
Locate in User	Select to enable searching for Provisioning Gateway user data under the Active Directory user objects.
Provide this setting for Sun LDAP storage only.	
User Prepend	Specifies the user naming attribute for user objects in the directory. This setting is used to form the relative distinguished name (RDN) of a user object. Typical values include "`CN`" or "`UID`."
Provide these settings for Oracle Directories, IBM LDAP Directory, Novell eDirectory, or Sun Directory Server storage only:	
*Connect as User	The user name of the directory administrator.
*Password	The password of the directory administrator.
Provide this setting for Oracle Directories, Active Directory, IBM LDAP Directory, Novell eDirectory, or Sun Directory Server storage only:	
Use secure connection (SSL)	Select to enable secure socket layer.
If using Configuration Objects or Role/Group support, provide these settings for all directory storage types:	
Use configuration objects instead of application list	Select to enable the use of configuration objects (COs) instead of application configuration lists, also known as the entlists. The Provisioning Gateway Server obtains the access control rights of its provisioning clients by searching the directory for provisioning objects. It finds only the objects to which it has access.

Setting	Value
Role/Group support	Select to enable Role/Group-based access control of administrative users. Enabling Role/Group support activates configuration object support. If Role/Group support is enabled, permissions should be specified. If no permissions are specified, by default, all users and groups are denied access for all actions. See Chapter 4, "Setting Up Role or Group Support" for information on setting up permissions.
Configuration and role/group objects root DN	Specifies where to begin the search for configuration and provisioning objects. The search moves from the specified locations downward. For example, `ou=vgoconfig,dc=test2003,dc=com or dc=passlogix,dc=com`. The path to this container must exist and contain at least one template prior to the input of these storage settings. The template can be in a sub-container rather than in the path itself. If this container does not exist, you will get an error message. See Chapter 4, "Setting Up Role or Group Support" for information on setting up permissions.

2.3 Event Log Settings

Use the **Event Log** page to configure the server where events will be logged. When you have completed your changes, click **Save Changes** to apply the new settings to Provisioning Gateway.

Setting	Description
Database Type	The Syslog Daemon is not a database; however, you select it on the Event Log Settings page from the Database Type drop-down list in order to send events to the daemon. There are no parameters to set for the Syslog daemon. Configuration is done manually following installation. See the *Oracle Enterprise Single Sign-On Suite Installation Guide* for more information.
Provide the following setting for Oracle Database only.	

Connection string	Enter the database connection string. For example, this string should be in the following form when using Oracle using external authentication:
	`Provider=[OLE DB Provider] ;Data Source=[SID]; User Id=/;`
	Microsoft's Oracle OLE Provider:
	`Provider=MSDAORA ;Data Source=ORCL; User Id=/;`

Provide the following setting for Microsoft SQL Server database only.	
Server	Enter the name of the server where events will be logged. SQL Server must be running on this machine, although the Provisioning Gateway database does not have to exist. If this is the first time this server is used by Provisioning Gateway, the Initialize Event Log box must also be checked to create the Provisioning Gateway database. You cannot use the IP address of the server to specify the current machine. You must use the actual machine name (for example, *pdevrx2*). You cannot use the name *localhost* to refer to the local machine. You must use the name of the machine.
Provide the following setting for the Oracle and SQL databases.	

Setting	Description
Initialize Event Log	When enabled, this setting creates the Provisioning Gateway database on the specified server. If the database already exists, all existing data in the database is erased. Typically, this setting is used for initial installation and when you want to clear the log entries in the database. This setting is not saved.

2.4 Template Mapping Settings

Use this page to map Logon Manager templates to Oracle Privileged Accounts Manager (OPAM) targets.

> **Note:** In order to perform any of the following functions, the user must be granted "Map Template" permissions in the Oracle Enterprise Single Sign-On Administrative Console.

1. In the **Targets** window, you will see the names of all available OPAM targets, followed by the name of the template mapped to it (in parentheses), if any.
2. Select a target and click the **Edit** button to edit the target's mapping properties.
3. In the template mapping **Edit** dialog, select a template to map to the OPAM target. If a template is already mapped to the target, it is selected when this dialog launches.

For more information about setting up template mapping and assigning permissions, see the *Oracle Enterprise Single Sign-On Suite Administrator's Guide*.

3 Managing Users

The Provisioning Gateway Administrative Console provides controls to configure user rights and activities. This chapter discusses the Console pages that contain these user settings.

3.1 Manage SSO Users Page

This page allows you to search for users and to add, modify, or delete their credentials. You can search for users by name or by their logons.

3.1.1 Finding Users

Use these parameters to specify the scope and specificity of a user search.

- **Show user(s) with User Name.** Enter the user name to search for. Leave this field blank to perform a search on all users. In the drop-down list, select either **substring match** or **exact match**.
- **Only show users who have logons for.** This list includes all the possible applications available to users in your organization. Select one or more application to filter the result to show only users who have logons for these applications.
- **Show additional information.** The search results list the usernames. The search results can also show **Logons** or **Pending Provisioning Instructions**. Select either of these options if desired.

Click **Find Users** after you have entered all necessary information.

3.1.2 Search Results

The results list the **User Name** and, depending on whether additional information was selected, **Logons** and, if applicable, any **Pending Provisioning Instructions**. Use the buttons (which highlight on mouse-over) to add, delete, and modify users. Click on a user's name to view or edit that user's profile.

> **Note:** You cannot provision applications that are not predefined (for example, on-the-fly Web applications).

➕	Add New Logon

☒	Delete SSO User or Delete Logon or Cancel Provisioning Request
↻	Modify Logon

3.1.2.1 Managing Users

The following settings provide control over SSO users.

3.1.2.1.1 Add New Logon This page allows you to create a provisioning instruction to add a new application logon for a specific user. This page is accessed by searching for a user on the Manage SSO Users page and clicking the button next to the **User Name**.

- **Add Logons**
  - **SSO User.** The Logon Manager user name selected from the user search results.
  - **Application.** Lists all of the available applications. There is also an option to **not list applications that user already has a logon for**. After an application is selected, the Logon Information section refreshes and text boxes appear for each field required by the selected application.
  - **Description.** Allows you to modify a logon's description field as seen in the Logon Manager Logon Manager. This field is optional.
- **Logon Information**
  - **User ID.** User's username or ID for the application.
  - **Password/Confirm Password.** User's password for the application.

> **Note:** After the **User ID** field is created, it cannot be modified. If a User ID must be changed, you must delete the existing logon and add a new logon with a new User ID. Depending on the requirements of the application being added, you might be prompted for additional fields, such as a Third or Fourth Field. Similarly, some applications might not require all of the fields. In such cases, the unnecessary fields do not appear.
>
> When you have entered all the required information, click **Add Logon** to submit your add request.

3.1.2.1.2 Delete SSO User This dialog asks if you are sure that you want to delete the selected SSO user. Click **OK** to delete or **Cancel** if do not want to delete this user. When you click **OK**, a message will confirm that this user has been deleted.

Access this dialog by searching for a user on the Manage SSO Users page and clicking the button next to **User Name**.

3.1.2.1.3 Delete Logon This dialog asks if you are sure that you want to delete the selected logon. Click **OK** to delete or **Cancel** if do not want to delete this logon. When you click **OK**, a message will confirm that this logon has been deleted. Access this dialog by searching for a user on the Manage SSO Users page and clicking the button next to **Logon**.

3.1.2.1.4 Cancel Request This dialog asks if you are sure that you want to cancel the pending provisioning instruction. Click **OK** to cancel or **Cancel** if do not want to cancel this request. When you click **OK**, the page will refresh and the pending provisioning instruction will no longer be displayed. Access this dialog by searching for a user on the Manage SSO Users page and clicking the button next to **Pending Provisioning Request**.

3.1.2.1.5 Modify Logon This page allows you to modify an application logon. Any fields that you leave blank on this page will not be changed. Access this page by searching for a user on the Manage SSP Users page and clicking the button next to **User Name**.

- **Logon to Modify**
  - **SSO User.** The Logon Manager user name selected from the user search results.
  - **Application.** The application to be modified.
  - **User ID.** Username or ID for the application.

> **Note:** After the User ID field is created, it cannot be modified. If a User ID must be changed, you must delete the existing logon and add a new logon with a new User ID.
>
> If a logon does not have User ID associated with it, the password field cannot be modified. A User ID must exist in order to modify the password. Logons that do not have a User ID associated with them should be deleted and recreated with a User ID, if a new one is required.

- **New Logon Information**
  - **Password/Confirm Password.** User's password for the application.
  - **Description.** Allows you to modify a logon's description field as seen in the Provisioning Gateway Logon Manager.
  - **Third Field.** The third field for this logon.
  - **Fourth Field.** The fourth field for this logon.

> **Note:** Third and Fourth Fields are required only if the identified application is configured with a Third or Fourth Field. Depending on the requirements of the application being added, you might be prompted for additional fields. Some applications might not require all of the fields. In such cases, the unnecessary fields do not appear.
>
> When you have entered all the necessary information, click **Modify Logon** to submit your modify request.

3.1.2.1.6 Edit User This page displays the selected user's logons and any pending provisioning instructions. Access this page by searching for a user on the Manage SSO Users page and clicking on the user's name in the search results list.

User Name	Displays the selected user's name.

➕	Click to add a new logon for this user.
✖	Click to delete this user.
Logons	Lists the logons assigned to the user.

Use the links and buttons (which highlight on mouse-over) to add, delete, and modify user logons.

Delete All Logons	Removes all logon credentials from the user's directory.
Advanced Delete	Allows you to generate a custom delete request.
✖	Deletes the specific logon associated with this user.
🔄	Changes a user's logon credentials for a specific logon.

> **Note:** If a logon does not have user ID associated with it, the password field cannot be modified. Any credentials that do not have a user ID associated with them should be deleted and replaced.

Pending Provisioning Items	Displays any provisioning instructions pending for the selected user. Displays the provisioning instruction (such as add or delete), the application, and the creation and execution date for the provisioning instruction. Click **Cancel Instruction** to delete this instruction from the repository.

3.1.2.1.7 Advanced Delete

- **SSO User.** Displays the Logon Manager user name selected from the user search results.
- **Application.** Lists the applications that can be deleted from this user. Select the application to delete from the drop-down list. The credential fields associated with the selected application are displayed. You must fill in all the credential fields exactly as they are stored in the directory:
  - **User ID.** Enter the User ID.
  - **Password/Confirm Password.** User's password for the application. These fields only appear if the application is configured to only have a password field.
  - **Description.** Logon's description field as seen in the Logon Manager Logon Manager.
  - **Third Field.** The third field for this application logon.
  - **Fourth Field.** The fourth field for this application logon.

When you have entered all the information has been entered, click **Submit** to submit your delete request.

3.1.2.1.8 Add New SSO User This page allows you to create new Logon Manager users. This creates a storage object in the repository for the user. After the user is created, the Add New Logon page appears so that you can add applications for the new user.

User Name. Enter the user name to add. Click **Next**. The Add New Logon screen opens.

Note: The user name must exist in the directory. If it does not, an error will occur.

4

Setting Up Role or Group Support

Provisioning Gateway Role/Group support provides the capability to manage provisioning rights for specific applications and users. These provisioning rights are configured and managed in the Oracle Enterprise Single Sign-On Administrative Console. To set up Role/Group support, open the Oracle Enterprise Single Sign-On Administrative Console by clicking **Start** > **Programs** > **Oracle**> **Logon Manager Console**.

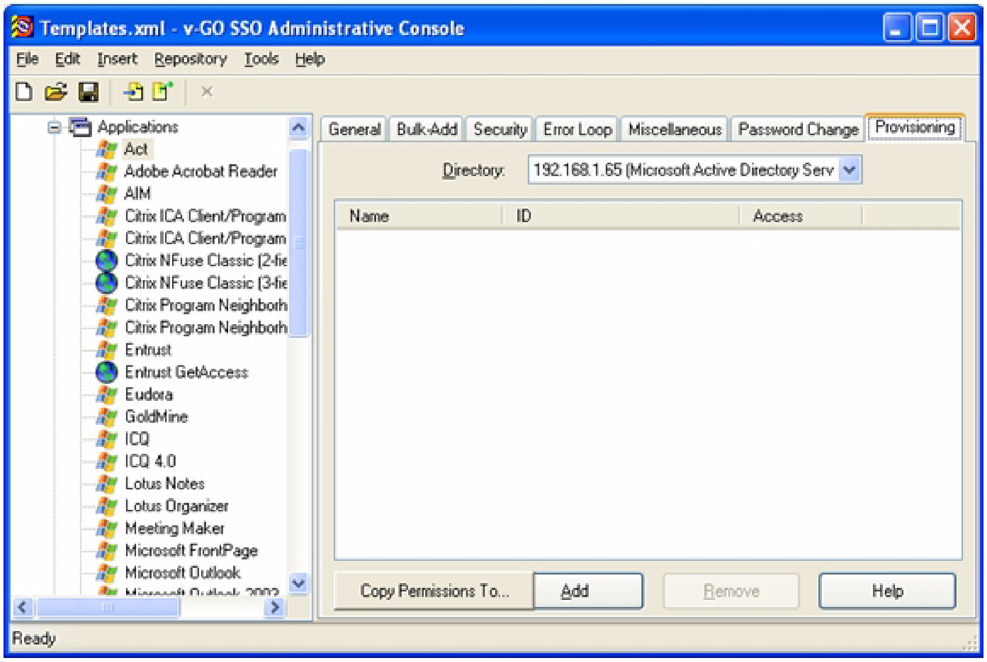
Two panels are available to manage provisioning rights:

- A **Provisioning** tab, which is located on the individual application panel. This tab enables you to manage provisioning rights for specific applications.
- A **Provisioning Manager** node, located in the Oracle Enterprise Single Sign-On Administrative Console tree (left pane). This node enables you to manage provisioning rights for users.

4.1 Using the Provisioning Tab

To access the **Provisioning** tab, expand **Applications** on the left side of the Oracle Enterprise Single Sign-On Administrative Console and double-click any application. Click the **Provisioning** tab.

From this tab, you can add or remove permissions. You can also select the level of access rights (add, modify, or delete applications) for those permissions.


Templates.xml - v-GO SSO Administrative Console
File Edit Insert Repository Tools Help
Applications
Act
Adobe Acrobat Reader
AIM
Citrix ICA Client/Program
Citrix ICA Client/Program
Citrix NFuse Classic (2-fie
Citrix NFuse Classic (3-fie
Citrix Program Neighborh
Citrix Program Neighborh
Entrust
Entrust GetAccess
Eudora
GoldMine
ICQ
ICQ 4.0
Lotus Notes
Lotus Organizer
Meeting Maker
Microsoft FrontPage
Microsoft Outlook
General Bulk-Add Security Error Loop Miscellaneous Password Change Provisioning
Name ID Access
Copy Permissions To... Add Remove Help
Ready

Directory: 192.168.1.65 (Microsoft Active Directory Serv

Control	Value
Directory	Enables you to select the target directory server.
Name	Lists the groups or users who currently have access to this item.
ID	Lists the user's account name.
Access	Indicates the permissions that have been granted to the user or group (Add, Modify or Delete Logon). To change a user or group's access rights, right-click the user or group and select **Add Logon**, **Modify Logon**, or **Delete Logon** from the shortcut menu.
Copy Permissions To	Enables you to apply the provisioning rights for the current application to multiple applications. Click this button to display a dialog listing all the applications. Select the applications that you want these provisioning rights to be copied to. Use **Ctrl+click** or **Shift+click** to select multiple entries. Click **OK**.
Remove	Removes selected users or groups from the list. Select a user or group to remove; use **Ctrl+click** or **Shift+click** to select multiple entries.

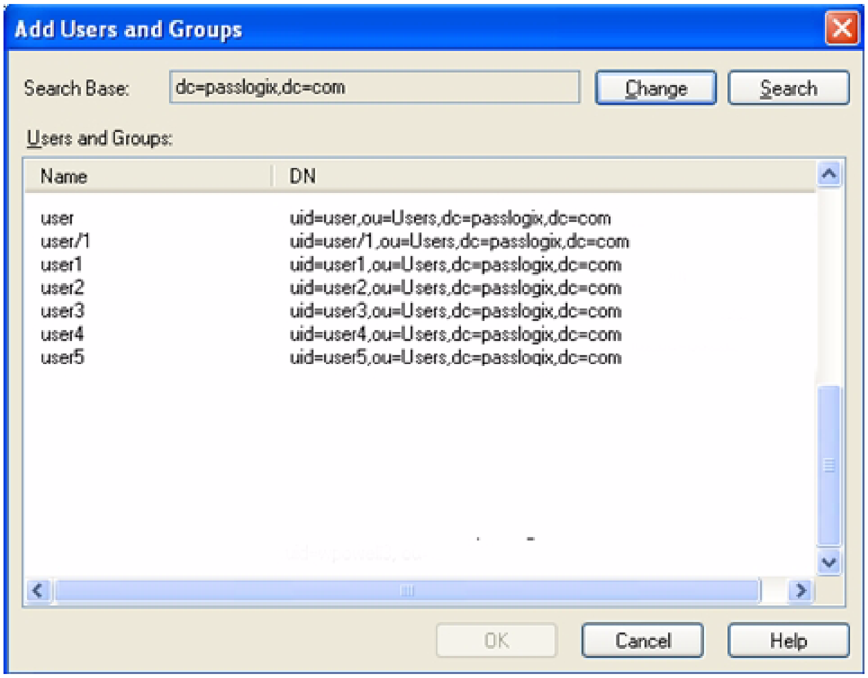
4.1.1 Adding Users or Groups

The dialog that you use to add users or groups depends upon which directory server is being used:

- LDAP
- Active Directory or AD LDS (ADAM)

4.1.1.1 LDAP

Use the Add Users and Groups dialog to select the individual users or user groups that are to be added to the access list for the current configuration item (Add Logon, Modify Logon, or Delete Logon).



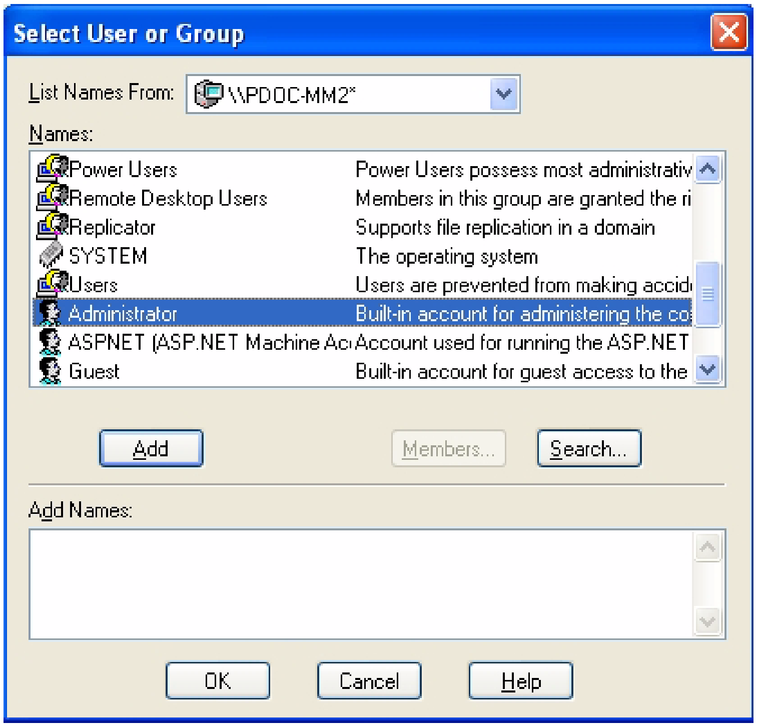
Add Users and Groups

Name	DN
user	uid=user,ou=Users,dc=passlogix,dc=com
user/1	uid=user/1,ou=Users,dc=passlogix,dc=com
user1	uid=user1,ou=Users,dc=passlogix,dc=com
user2	uid=user2,ou=Users,dc=passlogix,dc=com
user3	uid=user3,ou=Users,dc=passlogix,dc=com
user4	uid=user4,ou=Users,dc=passlogix,dc=com
user5	uid=user5,ou=Users,dc=passlogix,dc=com

Control	Value
Search Base	The base (highest-level) directory to begin searching for user or group accounts. All subdirectories of the base directory are searched. Enter a location or click **Change** to browse the directory tree.
Change	Displays the **Select Search Base** dialog to browse for a base directory for the search. Use this dialog to browse to and select the base (highest-level) directory to search for user and group names. Click **OK** when finished.
Search	Begin searching the base directory for users and groups.
Users or Groups	Lists the search results. Select the names to be added to the access list for the current configuration item. Use **Ctrl+click** or **Shift+click** to select multiple entries. Click **OK** when finished to copy your selections to the access list.

4.1.1.2 Active Directory and AD LDS (ADAM)

Use the Select User or Group dialog to select the individual users or user groups that are to be added to the access list for the current configuration item (Add Logon, Modify Logon, or Delete Logon).



List Names From: \\PDOC-MM2*

Power Users	Power Users possess most administrativ
Remote Desktop Users	Members in this group are granted the ri
Replicator	Supports file replication in a domain
SYSTEM	The operating system
Users	Users are prevented from making accid
Administrator	Built-in account for administering the co
ASPNET (ASP.NET Machine Ac	Account used for running the ASP.NET
Guest	Built-in account for guest access to the

Controls	Value
List Names From	Select an Active Directory domain or server.
Names	Lists the names of users and groups for the selected domain or server. Select one or more names to add to the access list.
Add	Copies users and groups selected in the **Names** list to the **Add Names** list. Use **Ctrl+click** or **Shift+click** to select multiple entries.
Members	When a group is selected the **Names** list, displays the **Global Group Membership** dialog, which lists the members of the selected group.
Search	When a group is selected the **Names** list, displays the **Global Group Membership** dialog, which lists the members of the selected group.
Add Names	Displays the names of the users or groups that you have you have added so far. Click **OK** to add these names to the access list for the current configuration item. You can type or edit user names in this list. However, entries are checked for invalid account names, and duplicate account selections are automatically removed when you click **OK**.

5

The Provisioning Gateway Node

Use the **Provisioning Gateway** node to manage provisioning rights for users. To access this functionality, click the **Provisioning Gateway** node from the tree in the left pane.

5.1 Provisioning Tabs

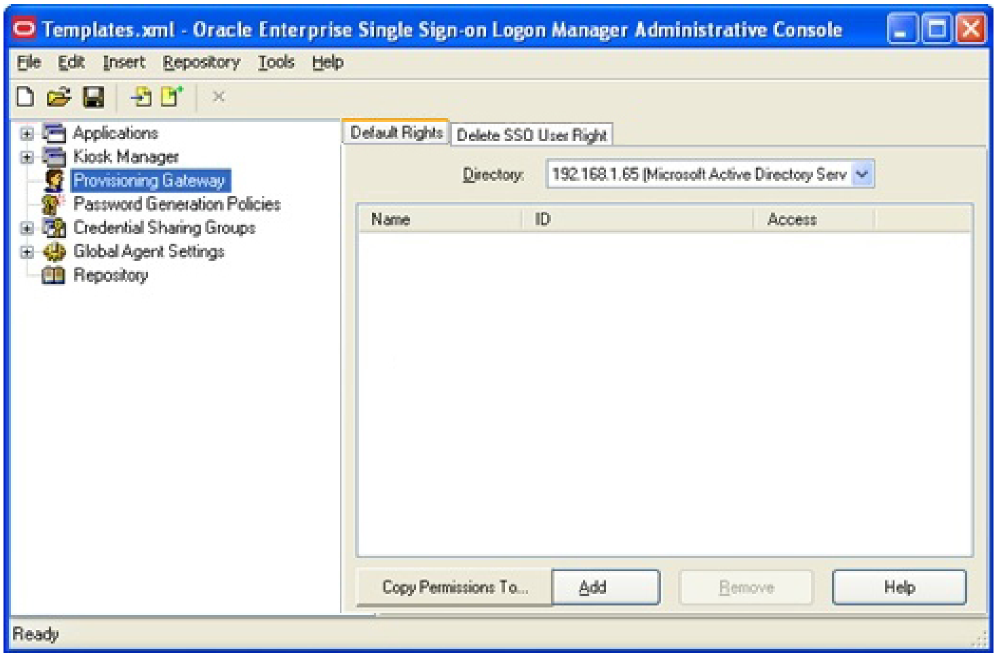
When you select the **Provisioning Gateway** node, the right pane is displayed with two tabs:

- Default Rights
- Delete SSO User Right

When you change the settings in this node, you must publish them to the repository in order for them to take effect. Right-click the **Provisioning Gateway** node in the Oracle Enterprise Single Sign-On Administrative Console, and select **Publish**.

5.1.1 Default Rights Tab

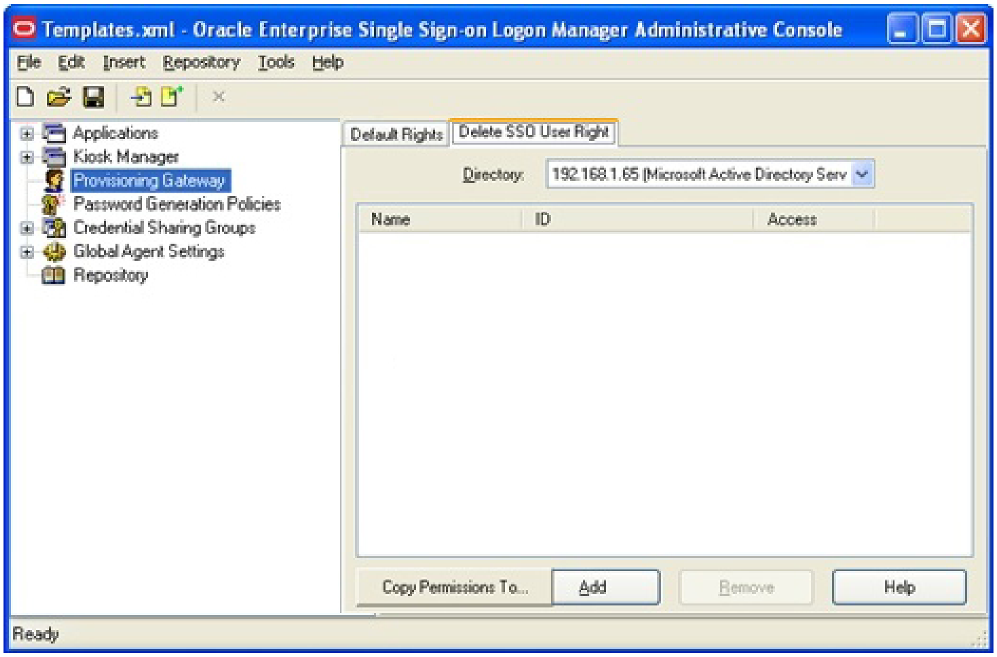
Use the **Defaults Rights** tab to define the provisioning rights for each new application created. This feature sets standard rights for each application created. After each application is created, change the rights as needed.



The controls on this tab function in the same manner as those on the **Provisioning** tab for an application.

5.1.2 Delete SSO User Right Tab

Use the **Delete SSO User Right** tab to define the users to grant the Delete SSO User functionality to in the Provisioning Gateway Management Console.


Templates.xml - Oracle Enterprise Single Sign-on Logon Manager Administrative Console
File Edit Insert Repository Tools Help
Applications
Kiosk Manager
Provisioning Gateway
Password Generation Policies
Credential Sharing Groups
Global Agent Settings
Repository
Default Rights
Delete SSO User Right
Name
ID
Access
Copy Permissions To...
Add
Remove
Help
Ready

Directory: 192.168.1.65 (Microsoft Active Directory Serv

The controls on this tab function in the same manner as those on the **Provisioning** tab for an application.

6

Reports and Logs

Use the Reports and Logs page to view Provisioning Gateway events. You can view events by date ranges and filter events by type.

6.1 Types of Reports

The Reports and Logs page has three options:

- Event Log
- Status Request
- Generate Report

6.1.1 Event Log

To configure an event log:

1. To select a date, click **Choose**.
2. Enter appropriate search parameters and click **View Log**. The log entries appear at the bottom of the screen:
   - Date/Time
   - Event Type
   - Provisioned User
   - Application
   - Execute Date
3. Click thebutton for details on the status of the instruction.

The log is exportable to a CSV file, which can be loaded into many optional tools (Microsoft Excel, for example) for analysis.

To export the log file:

1. To export the log file, click **Export Log**.
2. Select the export destination for the log file and click **OK**. These are the list of fields exported to this file:
   - Time Stamp
   - Event Type
   - User Name
   - Application

- Execute Date
- Provisioning Agent

6.1.2 Status Request

The Status page provides a summary of the status of the selected provisioning instruction.

- **State.** The state of the instruction:
  - Pending
  - Retrieved
  - Processed
- **Result.** The result of the instruction:
  - Success
  - Failure
  - Retrieved
- **Description.** A detailed textual description of the instruction processing result.
- **Modified Date.** The last time the instruction was modified. If the state of the instruction is "Pending," all the other fields are left blank.

Click **Back to Event Log** to return to the Event Log page.

6.1.3 Generate Report

Use the Generate Report page to download a CSV-formatted file containing all the data stored in the repository.

Select the type of report to generate:

- **Logons.** This option generates an application report (user's credentials). This report contains the following fields:
  - User DN (for example, cn=user1,ou=people,ou=vgo,dc=passlogix,dc=com)
  - User name (for example, user1)
  - Application Name
  - Last Used Date
  - Modified Date
- **Provisioning Instructions.** This option generates a provisioning item report (user's provisioning instructions). This report contains the following fields:
  - Instruction Type
  - Instruction GUID
  - Current Status
  - Provisioned User
  - Application
  - Create Date/Time
  - Execute Date/Time

  - Provisioning Agent

Select the type of report to generate and click **Download Report**.

7 The Provisioning Gateway CLIs

The Provisioning Gateway server exposes a Web service interface that allows it to receive instructions submitted to it by any other provisioning server. The Provisioning Gateway CLI is supplied as an integration component for provisioning solutions.

7.1 About CLIs

Provisioning Gateway includes two types of CLIs:

- .NET CLI

  The .NET CLI provides an interface for communicating with the Provisioning Gateway Web Service and is installed by default. The programming APIs are kept inside the Passlogix.Provisioning.dll assembly, which leverages the main .NET CLI executable as an SDK library. The .NET section of this guide is intended for experienced .NET application programmers responsible for the development of an organization's provisioning solutions.

- Java CLI

  The Java CLI exposes several interfaces, a class factory, and supporting types for communicating with the Provisioning Gateway Web Service. These programming APIs are kept inside the class library pmcli.jar, which is the same library that is the main executable for the Java CLI and is reused for the SDK. The Java section of this guide describes how to use the interfaces exposed by the Java CLI in your own applications.

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

For instructions to implement the .NET and Java CLIs, see *Developing Application Interfaces for Oracle Enterprise Single Sign-On Suite*.

> **Note:** The functionality of the .NET and Java CLIs is almost identical. The minor differences are noted throughout the document.

The .NET CLI is installed by default. To install the Java CLI, you must select the **Custom** option during installation.

7.2 CLI Syntax

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, in any combination:

Switch	Description
`-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`
`-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.

7.2.1 Differences Between .NET and Java CLI Approaches

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 `d` 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 exist 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 you use a properties file, you must set the value of `%P%` to refer to the name of the directory where you will place this file. This directory's name must not contain spaces.

7.3 Modes of Operation

There are three supported modes of operation:

- Command-Line Mode
- Batch Mode
- Interactive Mode

7.3.1 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.

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

7.3.2 Batch Mode

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

7.3.3 Interactive Mode

If there is no operation specified on the command line and no 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 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

7.3.4 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%).

Switch	Default
`-password`	The CLI will prompt for a password.

Note: Difference Between .NET and Java CLI

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

7.3.5 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 was 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 submitted successfully.	
`[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 descriptons for the information being returned.	
In the event of an error, the output will be the exception followed by a descriptive message, as follows: `[exception]: [descriptive error message]`		

7.3.6 Usage

The command `pmcli -?` displays usage and syntax information.

7.3.7 Status Results

When the Logon Manager Agent finishes 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

7.4 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).

7.4.1 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 Provisioning Gateway. This is the ID that the Provisioning Service uses to locate the user in the Provisioning Gateway 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.

7.4.2 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]
```

Note: 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.

7.4.3 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 Active Directory, 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.

7.4.3.1 Examples

The following examples demonstrate how to use the CLI.

7.4.3.1.1 Switches

```
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.

7.4.3.1.2 Smart Defaults

```
pmcli -p:Password
```

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

agent defaults to machine name

username is the current logged on 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 prompts for a password)

7.4.3.1.3 Adding a Credential

The following 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 Provisioning Gateway 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.

7.4.3.1.4 Deleting All Credentials for a User

The following 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
```

7.4.3.1.5 Returning a List of Specific Users 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
```

7.4.3.1.6 Executing Operations from a Batch File

The following 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 one provisioning operation per 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"
```

7.4.3.1.7 Running the CLI in Interactive Mode

The following 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 enters interactive mode and displays the following:

```
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 a session.

7.4.3.1.8 Displaying Help

```
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 he encounters a quit or exit.

7.4.3.1.9 Specifying When to Run the Provisioning Operation

The following 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 Logon Manager Agent should execute the provisioning operation only on or after the specified time. The operation exists on the directory service and the Provisioning Gateway Agent executes 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"
```

7.5 Using the .NET CLI as an SDK

The Provisioning Gateway .NET CLI must be installed prior to performing the steps in this section. Refer to the *Oracle Enterprise Single Sign-On Suite Installation Guide* for information on installing the Provisioning Gateway .NET CLI.

The .NET CLI is located under `<Passlogix home>\v-GO PM\Client\CLI\DotNet`.

To use the .NET CLI as an SDK, complete the following steps:

1. In your .NET project, add a reference to the Passlogix.Provisioning.dll.
2. Create an instance of the `IProvisioning` interface.
3. Call the available methods on this interface (such as `AddCredential`, etc).
4. Use the returned `IProvisioningResult` interface to determine success and retrieve results.

7.5.1 Adding a Reference to Passlogix.Provisioning

To add a reference to `Passlogix.Provisioning.dll` in your .NET project:

1. From Visual Studio, load your solution and launch the **Solution Explorer**.
2. Select the applicable .NET project and expand it.
3. Right click on the **References** node and select **Add Reference**.
4. From the dialog, select **Browse** and find `Passlogix.Provisioning.dll` (which you will find under `<Passlogix home>\v-GO PM\Client\DotNet`).
5. Click **Open**. A new reference to the assembly is created.
6. Open the source file (with `.cs` extension) where the APIs are called, and add the following lines at the beginning of the file:

   ```
   using Passlogix.Provisioning;

   using Passlogix.Provisioning.Exceptions;
   ```

7.5.2 Creating an Instance of the IProvisioning Interface

In the same file, create a method to initialize an instance of the `IProvisioning` interface and add one of the following lines to that method:

Method 1. If you know the full path

```
IProvisioning iprov =

ProvisioningFactory.CreateFrom(@"<Path to .NET CLI>");
```

Method 2. Load from same directory as provisioning assembly

```
IProvisioning iprov = ProvisioningFactory.CreateFromPrivate();
```

Method 3. To load file from the path (specified by %PATH%)

```
IProvisioning iprov = ProvisioningFactory.CreateFromPath();
```

After you have selected a method for loading, check for errors and then set the credentials for connection to the Provisioning Gateway service.

Use the following code after selecting the loading assembly method:

```
if (iprov != null)
```

```
{

try

{
```

You must first establish a connection to ensure that all resulting calls to the methods do not fail. This method sets credentials for connecting to the provisioning service. It does not actually connect to the service until a provisioning request is made.

There are three ways to connect:

Method 1

```
iprov.Connect("Administrator", "password");
```

Assumes `http://localhost/v-go pm service/up.asmx` and `%COMPUTERNAME%` is the Agent name.

Method 2

Specify the URL and Agent name:

```
iprov.Connect(

"http://<server>/v-go pm service/up.asmx",

"My Agent",

"Administrator", "password");
```

Method 3

Specify the URL:

```
iprov.Connect(

"http://<server>/v-go pm service/up.asmx",

"Administrator", "password");
```

Make provisioning requests via the `iprov` interface. This method is preferred because the Web service is not local but the user does not necessarily want to specify the agent name (defaults to `%COMPUTERNAME%`). See Sample Code for examples.

```
}

catch (ProvisioningException ex) { // Handle exception } }
```

After the connection executes successfully, requests can be sent to the Provisioning Gateway Web service through the methods of the `iprov` variable. Each method returns its results in an `IProvisioningnResult` interface. Oracle recommends these methods be called within a `try…catch` block for error handling. Catching the `ProvisioningException` class is sufficient for any exceptions thrown by the CLI. Other exceptions can be handled by adding a `catch (Exception)` block.

7.5.2.1 Available Methods in iProv Interface

This section lists all the available methods and their parameters for each provisioning operation. The following information is provided for each available method:

- Method name and description
- Method Overload List
- A description of the method's parameters (if applicable)

One of these parameters requires a special explanation. The `options` parameter is a dictionary of key-value pairs. The key is the name of the argument used by the CLI on the command line. The value is its value. The developer can set a key-value pair in the dictionary using either the literal name of the key (passed on the command line) or the key constants defined in the `OperationKeys` class.

- Command-line syntax used by the CLI (`CLI_Syntax`) (if applicable)

The command-line arguments map directly to the valid keys that can be used to fill the options parameter of a method. The `OperationKeys` class has been provided for convenience with constants mapping to the literal value of each key. This can be used to fill or index the options array. For brevity, the CLI Syntax does not show the full syntax. Refer to CLI Syntax for full information. The operation name is capitalized. Arguments specified in brackets are optional.

Method	Description
`Connect`	Establishes connection to Web service. This method does not actually attempt the connection but stores the credentials used to connect for use by other methods.

Overload List
`void Connect(string strUsername, string strPassword);`
`void Connect(string strURL, string strUsername, string strPassword);`
`void Connect(`
`string strURL,`
`string strAgent,`
`string strUsername,`
`string strPassword);`	

Parameter	Description
`strURL`	Web Service URL. Default is `http://localhost/v-GO%20PM%20Service/up.asmx`
`strAgent`	Identifier for this agent. Default is `%COMPUTERNAME%`.
`strUsername`	Username used to authenticate against the Web service.
`strPassword`	Password used to authenticate against the Web service.

Method	Description
`SetExecTime`	Sets the execution time of the provisioning instruction. This can be used to tell the instruction to execute in the agent at a future date or time after it has been created. If this is not set, it defaults to "Now."
Overload List void `SetExecTime(DateTime dtExec);`	

Method	Description
`AddCredential`	Provision the user with a new credential.

Method	Description
Overload List	
IProvisioningResult AddCredential(string strUserId, string strApplication, string strDescription, string strAppUserId, string strPassword); IProvisioningResult AddCredential(string strUserId, string strApplication, StringDictionary options);	

Parameter	Description
strUserId	User ID of user to be provisioned.
strApplication	Name of the application to provision.
strDescription	Description of the provisioning instruction.
strAppUserId	Application user ID of the credential.
strPassword	Password of the credential.
options	Hashtable of options (keys specified by OperationKeys).
CLI Syntax	
ADD_CREDENTIAL sso_userid sso_application [sso_app_userid] sso_password] [sso_description] [sso_other1] [sso_other2]	

Method	Description
CancelRequest	Cancel the provisioning request (before the agent runs).
Overload List	
IProvisioningResult CancelRequest(string strUserId, string strGuid);	

Parameter	Description
strUserId	User ID of user to be provisioned.
strGuid	ID of provisioning instruction to cancel (returned by several methods) that can be canceled.
CLI Syntax	
CANCEL sso_userid=<username> command_id=<guid>	

Method	Description
DeleteCredential	Delete a provisioned credential.

Method	Description
Overload List	
`IProvisioningResult DeleteCredential(string strUserId,`	
`string strApplication, string strAppUserId, string strOther1,`	
`string strOther2);`	
`IProvisioningResult DeleteCredential(string strUserId,`	
`string strApplication, StringDictionary options);`	

Parameter	Description
`strUserId`	User ID of user to be provisioned.
`strApplication`	Name of the application to provision.
`strAppUserId`	Application User ID of the credential.
`strOther1`	Other field value (1).
`strOther2`	Other field value (2).
`options`	Hashtable of options (keys specified by `OperationKeys`).
CLI Syntax	

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

Method	Description
ModifyCredential	Modify a provisioned credential.
Overload List	

`IProvisioningResult ModifyCredential(string strUserId, string strApplication,`
`string strAppUserId, string strDescription, string strPassword, string strOther1,`
`string strOther2);`	

`IProvisioningResult ModifyCredential(string strUserId, string strApplication,`
`string strAppUserId, StringDictionary options);`	

Parameter	Description
`strUserId`	User ID of user to modify.
`strApplication`	Name of the application of credential to modify.
`strAppUserId`	Application User ID of the credential to modify.
`strAppUserId`	Password of the credential to modify.
`strDescription`	Description of the provisioning instruction.
`strOther1`	Other field value (1).
`strOther2`	Other field value (2).
`options`	Hashtable of options (keys specified by OperationKeys).
CLI Syntax	

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

Method	Description
`DeleteUser`	Delete the user container (similar to deleting all credentials for a particular user).
Overload List	
`IProvisioningResult DeleteUser(string strUserId);`	

Parameter	Description
`strUserId`	User ID of container to delete.
CLI Syntax	
`DELETE_USER sso_userid=<username>`	

Method	Description
`GetStatus`	Ping the server. If it returns successfully without error, the server is functioning.
Overload List `IProvisioningResult GetStatus();`	
CLI Syntax	
`CHECK_SERVER`	

Method	Description
`StatusRequest`	Request the status of a pending provisioning instruction.
Overload List `IProvisioningResult StatusRequest(string strUserId, string strGuid);`	

Parameter	Description
`strUserId`	User ID to query.
`strGuid`	ID of provisioning instruction (returned by several methods)
CLI Syntax	
`STATUS sso_userid=<username> command_id=<guid>`	

Method	Description
`GetSettings`	Return the directory settings of the PM Web service.
Overload List `IProvisioningResult GetSettings();`	
CLI Syntax	
`GET_SETTINGS`	

Method	Description
`GetSchema`	Get the schema (or list of available options for SetSettings).

Method	Description
Overload List	
`IProvisioningResult GetSchema();`	
CLI Syntax	
`GET_SCHEMA`	

Method	Description
`SetSettings`	Change the settings used by the Web service.
Overload List	
`IProvisioningResult SetSettings(IDictionary map);`	

Parameter	Description
`Map`	Key-value pair for each setting.
CLI Syntax	
`SET_SETTINGS name="key1, key2, ..." value="value1, value2, ..."`	

Method	Description
`ExtSearch`	Search the directory service and return information on users, applications, and logs. This returns a list of applications that can be provisioned for a particular user or all users.
Overload List for Applications	

`IProvisioningResult ExtSearchApplications();`
`IProvisioningResult ExtSearchApplications(string strUserId);`	

Parameter	Description
`strUserId`	Name of user whose application list should be returned.
Overload List for Users	

`IProvisioningResult ExtSearchUsers();`
`IProvisioningResult ExtSearchUsers(string strUserId,`
`StringCollection logons, bool fRetLogons, bool fRetInsts,`
`bool fMatchExact);`
`IProvisioningResult ExtSearchUsers(StringDictionary options);`	

Parameter	Description
`strUserId`	User to return information on.
`logons`	Return only these logons (csv format).
`fRetLogons`	Return logon information.
`fRetInsts`	Return pending provisioning instructions.
`fMatchExact`	Use exact match on `strUserId`.
`options`	Hashtable of options (specified by `ExtSearchKeys`).

Parameter	Description
Overload List for Logging	

`IProvisioningResult ExtSearchLog();`
`IProvisioningResult ExtSearchLog(EventType evt);`
`IProvisioningResult ExtSearchLog(DateTime dtStart, DateTime dtEnd,`
`EventType evt);`	

Parameter	Description
`evt`	`EventType` to return.
`dtStart`	Start date of range to return.
`dtEnd`	End date of range to return.
CLI Syntax	

`EXT_SEARCH CATALOG=Applications [userId=<username>]`
`EXT_SEARCH CATALOG=Users [userId=<username>]`
`[logon="logon1,logon2,..."] [returnLogons=true|false]`
`[returnInstructions=true|false] [uidMatch=substring|equal]`
`EXT_SEARCH CATALOG=EventLog [startDate=mm/dd/yyyy] [endDate=mm/dd/yyyy]`
`[eventType=amducs]`	

7.5.3 Retrieving Results

After a provisioning request to the Provisioning Gateway Web Service has completed, an `IProvisioningResult` interface is returned by the called method. Your application can use this interface to determine if the request has completed successfully and retrieve any relevant results. This section shows the available properties on the `IProvisioningResult` interface and how to interpret their values for the methods called from IProvisioning.

7.5.3.1 Interface Definition

```
public interface IProvisioningResult { string Response { get; }

bool Success { get; }

string CommandID { get; }

string ErrorMessage { get; }

IDictionary AttributesCollection { get; } }
```

Property	Description
`Success`	True if the command completed successfully.
`ErrorMessage`	The error string if `Success` is `False`. May not always be set.

Property	Description
CommandID	The unique ID associated with the completed command (a 32-digit GUID)). All methods except ExtSearch return a GUID. However, only the following methods provide a GUID that can be used by the CancelRequest and StatusRequest operation: ■ AddCredential ■ ModifyCredential ■ DeleteCredential
Response	The raw XML response returned by Web service. This is useful if the results need to be re-parsed.
AttributesCollection	Detailed results returned by the Web service on Success. The format is a Dictionary of key-value pairs. The methods that fill this property are: ■ GetSettings ■ GetSchema ■ StatusRequest ■ ExtSearch

7.5.4 Attributes Collection

The Attributes Collection is a dictionary of attributes returned by GetSettings, GetSchema, ExtSearch, and StatusRequest. The keys are strings that represent the attribute name. The values can refer either to another IDictionary, an IList, or a string. However, types are not mixed within the same collection. After the type is established, the same type is referenced by all keys.

The following table lists the keys and values returned by the provisioning operations and their meanings:

Methods	Description	
GetSettings	Returns a collection of string key-value pairs. The key is the name of the setting. The value is its value. These are the storage values set in the registry by the Provisioning Gateway Web Service.	
StatusRequest	Returns a collection of string key-value pairs. The key is the name of a status property. The value is its value. The following status keys are supported:	
	Status Key	Value
	InstructionState	PENDING, PROCESSED
	Result	SUCCESS, FAILED
	Description	SUCCESS, <Reason for failure>
	Modified	<Date modified>
GetSchema	The key is a string that represents the name of a group of storage settings. The value is an IList. Each IList entry describes one setting under this group. The entry is an IDictionary of string key-value pairs. The key can be one of the following followed by one of the possible values:	
	Key	Value
	DataType	Can be string or bool
	DisplayDesc	A description of this setting. Can be empty.
	DisplayName	The friendly name of this setting to display.

Methods	Description	
	`Flags`	An internal value used to describe if the settings is non-persistent, must exist.
	`RegDefault`	The default value for this setting. Can be empty.
	`RegName`	The name of the registry key.
	`RegPath`	The relative registry path to this setting.
	`RegType`	The registry type (`DWORD` or `string`).
	Note: The setting described by this entry becomes a value that can be retrieved or set by `GetSettings` and `SetSettings`.	
`ExtSearch`	Collection of hashtables. (See next section for more information). The key is a string but the type of the returned value depends on the `ExtSearchXXX` called.	

Note: The structure and format of the returned key-value pairs from the AttributesCollection property are designed to closely mirror the console output from the actual CLI. Simply using the CLI will help in understanding the format and structure of the collection returned by these methods.

7.5.5 ExtSearch Results

This section describes the format of the AttributesCollection map returned by ExtSearch.

7.5.5.1 ExtSearchApplications

Returns:	**.NET:** HashTable of HashTables **Java:** HashMap of HashMaps

Key	Value	
Application Name	HashTable (string key/value pairs)	
	Key	**Value**
	`HasFourthField`	`True | False`
	`HasPassword`	`True | False`
	`HasThirdField`	`True | False`
	`HasUserId`	`True | False`
	`IsSecurId`	`True | False`
	If `IsSecurId` is true, then the first four fields are renamed: ■ `SecurID-UserId` ■ `SecurID-Other[4th]` ■ `HasPassword` ■ `PassKeyType`	

The following are sample search results:

Adobe Acrobat Readers

```
HasFourthField: False

HasPassword: True

HasThirdField: False

IsSecurID: False

HasUserId: False
```

MSN Messenger

```
HasFourthField: False

HasPassword: True

HasThirdField: False

IsSecurID: False

HasUserId: True
```

Visual SourceSafe

```
HasFourthField: False

HasPassword: True

HasThirdField: True

IsSecurID: False

HasUserId: True
```

7.5.5.2 ExtSearchUsers

Returns:	**.NET:** `HashTable` of Lists of HashTables **Java:** `HashMap` of Lisis of HashMaps

Key	Value	
User's Name		
	Logon Entry	
	Key	**Value**
	`name`	Application name
	`modifiedDate`	Date last modified
	`lastUsedDate`	Date last used by SSO
	`Id`	GUID identifier
	Pending Entry	
	`applicationName`	Application
	`createDate`	Date created
	`executeDate`	Date this will execute
	`id`	GUID identifier

Key	Value	
	`instructionType`	`ADD | MODIFY | DELETE`
	`provisioningAgent`	Agent name
	`status`	`SUCCESS | Pending`

7.5.5.2.1 CLI Output

```
ext_search catalog=users returnLogons=true
```

This returns a list of logons for all users.

johnd

```
modifiedDate: 2005-08-24 16:43:41Z

lastUsedDate: 2005-08-24 16:43:41Z

name: Adobe Acrobat Reader

id: a75f58c8-a3bd-4d00-bc27-99a587dd98f8


modifiedDate: 2005-08-24 16:43:41Z

lastUsedDate: 2005-08-24 16:43:41Z

name: Adobe Acrobat Reader

id: d6bc375d-3f90-400b-a012-6b80aff4ef49


modifiedDate: 2005-09-09 16:28:15Z

lastUsedDate: 2005-09-09 16:28:15Z

name: Visual SourceSafe

id: 80cdc929-61a6-4b86-8763-d5f02b0dbb8b


modifiedDate: 2005-09-01 17:30:26Z

lastUsedDate: 2005-09-01 17:30:26Z

name: Visual SourceSafe

id: 065f5cff-b651-4a3a-a99c-c606059cbad7


modifiedDate: 2005-09-09 16:41:33Z

lastUsedDate: 2005-09-09 16:41:33Z

name: Visual SourceSafe

id: 0a0686b5-3e38-4830-8e02-79b8177de0b4
```

7.5.5.3 ExtSearchLog

Returns:	**.NET:** HashTable of HashTables **Java:** HashMap of HashMaps

Key	Value	
Entry Number	HashTable (string key/value pairs)	
	Key	**Value**
	`applicationName`	Application name
	`eventType`	Type of event (DWORD flag)
	`executeDate`	Date executed
	`id`	GUID identifier
	`provisionedUser`	User provisioned
	`provisioningAgent`	Agent name
	`timeStamp`	Time stamp

7.5.5.3.1 CLI Output

```
ext_search catalog=eventLog
```

This returns a list of logons for all users.

Entry 1

```
applicationName:
eventType: 64
executeDate: 0001-01-01 00:00:00.000Z
id: a09b9de7-4b65-464c-8dcb-90219e222991
provisionedUser:
provisioningAgent: SSO PM Console
timestamp: 2005-11-17 18:33:37.290Z
```

Entry 2

```
applicationName:
eventType: 64
executeDate: 0001-01-01 00:00:00.000Z
id: bd444f6c-e3cf-4efc-bbd8-c5e82d55ed96
provisionedUser:
provisioningAgent: SSO PM Console
timestamp: 2005-11-17 18:33:37.370Z
```

Entry 3

```
applicationName:
eventType: 64
executeDate: 0001-01-01 00:00:00.000Z
id: 6eebd1dd-a904-43db-8c22-38ef941e83b3
provisionedUser:
provisioningAgent: SSO PM Console
```

```
timestamp: 2005-11-17 18:33:38.960Z
```

Entry 4

```
applicationName: Visual SourceSafe

eventType: 4

executeDate: 2005-11-17 19:28:51.427Z

id: 2c45f078-c9c7-4268-9abd-4e50111ba644

provisionedUser: davidh

provisioningAgent: SSO PM Console

timestamp: 2005-11-17 19:28:51.427Z
```

7.5.6 Sample Code

The following code demonstrates how to call the `AddCredential` method from the `IProvisioning` interface. This example demonstrates adding a credential for the Logon Manager user `johndoe`. The application being added is `Yahoo` and the credentials for this application are `jdoe` and `password`.. The description of this credential is `TestApp`.

```
try

{

IProvisioningResult ipr = iprov.AddCredential(

"johndoe",

"Yahoo",

"Test App",

"jdoe",

"password");

//Process results in ipr

if (!ipr.Success)

{

Console.WriteLine(ipr.ErrorMessage);

return;

}

//Display GUID

Console.WriteLine("SUCCESS" + ipr.CommandID);

}

catch (ProvisioningException ex)

{

// Handle Exception...

}
```

Credentials can also be added using an options argument, which is a more flexible method of passing. This method allows the use of additional parameters (some applications require an `OTHER1` and `OTHER2` field) and their combinations.

The following example demonstrates how to add a credential for the `Visual SourceSafe` application for the SSO user `johndoe`. Since this application requires an `OTHER1` field, this method is the only way to add the credential.

```
StringDictionary options = new StringDictionary();

options.Add(OperationKeys.DESCRIPTION, "Test App");

options.Add(OperationKeys.APP_USERID, "jdoe");

options.Add(OperationKeys.PASSWORD, "password");

options.Add(OperationKeys.OTHER1, "VGO");

IProvisioningResult ipr = iprov.AddCredential("johndoe",

"Visual SourceSafe", options);
```

7.6 Using the Java CLI as an SDK

The Provisioning Gateway CLI must be installed prior to performing the steps in this section. Refer to the *Oracle Enterprise Single Sign-On Suite Installation and Setup Guide* for information on installing the Provisioning Gateway CLI.

The Java CLI is located under `<home directory>\v-GO PM\Client\Java\<version>`.

To use the Java CLI as an SDK, follow these steps:

- Add `pmcli.jar` and supporting libraries to the `CLASSPATH`.
- Import the provisioning classes into your application.
- Create an instance of the `ProvisioningConnection` class.
- Create an instance of the `CLIOperationParser` class.
- Define the operation and operation parameters using a `StringMap`.
- Create an instance of the Operation using the object instance created in step 4.
- Set execution time (otherwise it defaults to "Now").
- Send Operation instance (step 6) to the Web service using the `ProvisioningConnection` (step 3) instance.
- Retrieve success and results of operation.

7.6.1 Sample Code

The following code illustrates a simple program that implements each of these steps:

1. Import these classes into your application:

   ```
   import com.passlogix.vgo.pm.cli.*; import
   com.passlogix.vgo.pm.operations.*;
   ```

   Sample routine for calling the web service:

   ```
   void CallWebService(/* Parameters */) {
   ```

   Arguments to `ProvisioningConnection` are defined as:

`URL`: the webservice URL `strAgent`: the user-defined name for the client agent `strUsername`: the username to connect as `strPassword`: the password to use for connection `ProvisioningConnection conn = new ProvisioningConnection(strURL, strAgent, strUsername, strPassword); try {`

2. Begin execution of instruction:

   ```
   CLIOperationParser opParser = CLIOperationParser.newInstance();
   Operation.StringMap options = new Operation.StringMap();
   ```

3. Use `OperationKeys` class for most options. Use `ExtSearchKeys` class for `ExtSearch` operation:

   ```
   options[OperationKeys.USERID] = "davidh";
   options[OperationKeys.APPLICATION] = "Visual SourceSafe";
   ```

   `strOper` can be equal to any operation defined in `CLIOperationParser`: `Operation oper = opParser.parse(strOper, options);`

4. Set the execution time of instruction. If you leave the execution time unspecified, it defaults to `Now`. `oper.setExecTime(dtExec); conn.sendInstruction(oper);`

5. Get results if the operation was successful: `if (!oper.getSuccess()) { String strMsg = String.format(`

   ```
   "The command failed: id=%s, msg=%s", oper.getCommandID(),
   oper.getError()); return; }
   ```

6. Retrieve command ID and result attributes:

   String `strID = oper.getCommandID()); CollectionsMap map = oper.getResultAttributes()); } catch (Exception ex) { // print exception } }`

   For some commands, one or both of these is empty. See the section Using the .NET CLI as an SDK for more information on the command ID and format of result attributes and the available options for each operation. The available operations are defined as static members of the CLIOperationParser class. All of the available options and parameters for the supported operations are defined in OperationKeys Interface and ExtSearchKeys Interface sections of this document.

7.6.2 Class Definitions

The following class definitions show the important constants and methods needed to programmatically send a request to the Provisioning Gateway Web Service.

7.6.2.1 CLIOperationParser Class

This class inherits from `OperationParser`. An instance of itself can be created by calling newInstance(). When an instance exists, it can be used to create Operation objects representing the specific request to be executed on the server:

Following are all supported operations defined as constant strings:

```
static public final String ADD_CREDENTIAL = "add_credential";

static public final String MODIFY_CREDENTIAL = "modify_credential";

static public final String DELETE_CREDENTIAL = "delete_credential";

static public final String DELETE_USER = "delete_user";

static public final String STATUS = "status";
```

```
static public final String CANCEL = "cancel";

static public final String CHECK_SERVER = "check_server";

static public final String GET_SETTINGS = "get_settings";

static public final String GET_SCHEMA = "get_schema";

static public final String SET_SETTINGS = "set_settings";

static public final String EXT_SEARCH = "ext_search";
```

To create a new instance of this parser

```
static public CLIOperationParser newInstance();
```

To print the results to an output stream of choice

```
public void printResults(PrintStream out, Operation oper);
```

7.6.2.2 OperationParser Class

This class is the base class for `CLIOperationParser`. It defines methods for supporting additional operations and creating Operation objects:

- To add a new provisioning operation and its support class:

  ```
  public void addOperation(String strOper, Class<? extends Operation> c)
  ```

  The supporting class must be derived from the abstract `Operation` class. This method is intended for internal use.

- To create an instance of the `Operation` object for the given provisioning instruction:

  ```
  public Operation parse(String strInstr) throws InstantiationException,
  IllegalAccessException
  ```

  This instruction follows the same format as that passed in the command line.

- To create an instance of the `Operation` object based on the operation name:

  ```
  public Operation parseNoOpt(String strOper) throws
  InstantiationException, IllegalAccessException
  ```

- To create an instance of the `Operation` object based on the operation name and its parameters (specified as a map of key/value pairs):

  ```
  public Operation parse(String strOper, Operation.StringMap options)
  throws InstantiationException, IllegalAccessException
  ```

7.6.2.3 Operation Class

The Operation Class is the base class for all Operations supported by the Java CLI. This class is responsible for constructing the correct message to send to the Web service and for retrieving and storing the response. The following methods can be used to query the results:

- Get the raw xml response from the server:

  ```
  public String getResponse()
  ```

- Query if the operation executed successfully:

  ```
  public boolean getSuccess()
  ```

- Get the GUID associated with this operation after it is executed:

  ```
  public String getCommandID()
  ```

This can be an empty string if no GUID is associated with the operation.

- Get any error message if `getSuccess` returns false:

  ```
  public String getError()
  ```

- Set the execution time of this operation on the server:

  public void `setExecTime(Date dtExec)`

  If you do not provide a value for this parameter, the Operation executes immediately. Otherwise the Operation does not execute until the given time.

- Get the result attributes array if the operation was successful:

  ```
  public CollectionsMap getResultAttributes()
  ```

  An empty `CollectionsMap` cab be returned if there are no results other than success to return. The format of `CollectionsMap` is a name/value pair map of lists or other maps. The exact format of which depends on the operation executed. For more information, see Using the .NET CLI as an SDK.

- Execute the operation:

  ```
  public String send(ProvisioningConnection conn) throws PMCLIException,
  RemoteException
  ```

  You generally should not call this method directly. Instead call:

  ```
  ProvisioningConnection.sendInstruction(...)
  ```

  This passes the Operation object to it.

7.6.2.4 OperationKeys Interface

The OperationKeys interface defines all the possible parameters that an Operation can accept. The parameters are specified as keys to the `StringMap`, followed by their value. The exact subset of keys an Operation supports is described in the Provisioning Gateway Using the .NET CLI as an SDK.

```
public interface OperationKeys

{

static public final String USERID = "sso_userid";

static public final String APPLICATION = "sso_application";

static public final String DESCRIPTION = "sso_description";

static public final String APP_USERID = "sso_app_userid";

static public final String PASSWORD = "sso_password";

static public final String OTHER1 = "sso_other1";

static public final String OTHER2 = "sso_other2";

static public final String GUID = "command_id";

static public final String NAME = "name";

static public final String VALUE = "value"; }
```

7.6.2.5 ExtSearchKeys Interface

The `ExtSearchKeys` interface defines the parameters supported for the `ExtSearch` operation. The `OperationKeys` interface does not apply for this operation. Acceptable parameters must come from this list:

```
public interface ExtSearchKeys

{
```

Supported keys for ExtSearch

```
static public final String OPTION_CATALOG = "catalog";

static public final String OPTION_USERID = "userId";

static public final String OPTION_APPLICATION= "applicationName";

static public final String OPTION_EVENTTYPE = "eventType";

static public final String OPTION_STARTDATE = "startDate";

static public final String OPTION_ENDDATE = "endDate";

static public final String OPTION_LOGON = "logon";

static public final String OPTION_SHOWLOGONS = "returnlogons";

static public final String OPTION_SHOWPENDING = "returnInstructions";

static public final String OPTION_UIDMATCH = "uidMatch";
```

Possible values for OPTION_UIDMATCH key

```
static public final String MATCH_EQUAL = "equal";

static public final String MATCH_SUBSTRING = "substring";
```

Possible values for OPTION_CATALOG key

```
static public final String CATALOG_APPS = "Applications";

static public final String CATALOG_EVENTLOG = "EventLog";

static public final String CATALOG_USERS = "Users"; }
```

7.6.3 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.jar` 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`.
6. Save the file.
7. Add the full path to the directory where `pmcli.properties` lives (not the full path to the file) to the `CLASSPATH`.
8. Run `pmcli.bat` and pass an `https` URL to the `-url` switch.

> **Note:** Enabling SSL does not prevent the CLI from communicating with an http service.