Oracle® Enterprise Single Sign-On Suite Plus

Secure Deployment Guide

Release 11.1.2

E27159-02

August 2012

ORACLE

Oracle Enterprise Single Sign-On Suite Plus Secure Deployment Guide

Release 11.1.2

E27159-02

Table of Contents

Preface

Audience

This document describes how to securely deploy each component of the Oracle Enterprise Single Sign-On Suite Plus. It also advises against deployment scenarios that might arise that can adversely affect the security of your Oracle software deployment.

> **Warning:** For maximum security of Oracle and other applications, Oracle urges you to follow Oracle, Microsoft, and other network security best practices in your enterprise. This includes enforcing the use of strong passwords and policies across your network, including those used to authenticate to Oracle ESSO Suite applications, to reduce the potential for unauthorized access to sensitive information. Oracle also highly recommends minimizing the privileges granted to domain accounts for both users and applications, including disallowing local administrator access for all but those users that explicitly require it.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support.

For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

We continually strive to keep our documentation accurate and up to date. For the latest version of this and other documents, visit http://download.oracle.com/docs/cd/E23906_01/index.htm.

For more information, see the other documents in the documentation set for this release:

- **Oracle Enterprise Single Sign-On Suite Plus**
  - *Release Notes*
  - *Installation Guide*
  - *Administrator's Guide*
  - *Secure Deployment Guide*
  - *User's Guide*
- **Oracle Enterprise Single Sign-On Logon Manager**
  - *Deploying Logon Manager with Microsoft Active Directory*
  - *Deploying Logon Manager with Microsoft Active Directory Application Mode and Active Directory Lightweight Directory Services*
  - *Deploying Logon Manager with a Lightweight Directory Access Protocol Directory*
  - *Template Configuration and Diagnostics for Windows Applications*

    - *Template Configuration and Diagnostics for Web Applications*
    - *Template Configuration and Diagnostics for Mainframe Applications*
- **Oracle Enterprise Single Sign-On Provisioning Gateway**
    - *Administrator's Guide*
    - *Command Line Interface Guide*
    - *Oracle Identity Manager Connector Guide*
    - *Sun Java System Identity Manager Connector Guide*
    - *IBM Tivoli Identity Manager Connector Guide*
- **Oracle Enterprise Single Sign-On Universal Authentication Manager**
    - *Administrator's Guide*
    - *User's Guide*

Conventions

The following text conventions are used in this document:

Term or Abbreviation	Description
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.

Securing Logon Manager

When planning to deploy Logon Manager, review the following topics and follow the guidelines they describe to ensure a secure deployment:

- Deploying Logon Manager in Directory Environments
- Designing the Logon Manager Directory Sub-Tree
- Achieving a Secure Configuration of Logon Manager
- Deploying Logon Manager with the Windows Authenticator Version 2
- Ensuring Secure Response to Applications

Deploying Logon Manager in Directory Environments

You have the choice to deploy Logon Manager in a directory environment, such as Active Directory, ADAM/AD-LDS, Oracle Internet Directory, and many third-party LDAP-compliant directories, which enables the delivery of single sign-on capability to any machine on the network through central storage of application credentials, templates, and policies. Users synchronize with the directory to download these items and update their credential stores with new or changed usernames and passwords.

Adding Logon Manager to your existing directory environment provides the following benefits:

- Logon Manager leverages the existing user accounts, groups, and native directory permissions (ACLs) without the need to manage these items separately or synchronize them with another directory or database.
- Logon Manager data is automatically protected by your existing backup, failover, and disaster recovery plans.
- No dedicated servers or server-side processes are required; Logon Manager's scalability and performance depend solely on the capacity and robustness of your existing directory infrastructure.
- Administrators are not burdened with additional administrative tasks or the need to learn new tools or concepts. Delegated administration of Logon Manager is achieved through the native capabilities of the directory.

A directory also enables the organization of Logon Manager templates and policies into a highly visual hierarchy. While you can use a flat model if your environment calls for it, a properly set-up hierarchy can help maintain top directory, Agent, and network performance, as well as simplify Logon Manager administration and improve data security by permitting more efficient access control.

How Logon Manager Extends Your Active Directory Schema

Before Logon Manager can store data in Active Directory, you must instruct Logon Manager to extend your Active Directory schema. The schema extension consists of adding four object classes and setting the appropriate permissions so that objects of those types can be created, read, modified, and deleted. Existing classes and attributes are **not** modified in any way. If you decide to allow Logon Manager to

store application credentials under user objects (a recommended best practice), Logon Manager will also apply the permissions required by this feature.

How Logon Manager Synchronizes with Active Directory

The Logon Manager Agent uses the Active Directory synchronizer plug-in to communicate with Active Directory. When properly configured, synchronization occurs whenever one of the following events takes place:

- The Logon Manager Agent starts.
- Application credentials are added, modified, or deleted by the end-user.
- The machine running the Agent acquires an IP address or its existing IP address changes. (if Logon Manager is configured to respond to these events).
- The auto-synchronize interval elapses (if configured).
- The user initiates synchronization via Logon Manager's "Refresh" function.
- A script initiates synchronization using a command-line parameter.

During synchronization, the Logon Manager Agent traverses the Logon Manager tree and loads the contents of the sub-containers to which the current user has been granted access; it also synchronizes any credentials that have been added, modified, or deleted since the last synchronization.

How Logon Manager Handles and Stores Application Credentials

Logon Manager encrypts application credentials using a unique key generated when the user completes the First-Time Use (FTU) wizard. The credentials remain encrypted at all times, including in the Agent's local cache, the directory, and while in transit over the network. Logon Manager only decrypts credentials (to memory, never to disk) when a configured application requests logon, and wipes the target memory location as soon as the logon request completes. The amount of data Logon Manager stores per enabled application and per user is trivial (measurable in bytes and small kilobytes).

Further Reading

An in-depth discussion of the Logon Manager software architecture is beyond the scope of this guide. To obtain Oracle white papers containing additional information, contact your Oracle representative.

Designing the Logon Manager Directory Sub-Tree

Logon Manager gives you the freedom to set up the directory structure to best fit the needs of your organization. Specifically, you have the choice to store your data in a flat model, or create a hierarchy. While a flat model works fine for small deployments, growing and large deployments should utilize a hierarchy from the very beginning. The exact structure of your sub-tree will depend on the following factors:

- Number of users
- Number of applications you want Logon Manager to support
- Robustness of the existing infrastructure
- Structure of your organization

Always follow Microsoft's best practices for Active Directory design and implementation described in the following article: http://technet.microsoft.com/en-us/library/bb727085.aspx

Guidelines for Structuring the Sub-Tree

Oracle recommends that you set up your sub-tree as a hierarchy by following the guidelines below:

- Use OUs to group templates and policies by category, such as department or division, according to the structure of your organization.
- Control access at the OU level.
- Disable inheritance and grant no user rights at the root of the Logon Manager sub-tree, unless your environment dictates otherwise.

When set up this way, a hierarchy provides the following benefits:

- **Highly visual and self-documenting tree structure.** When you view your sub-tree in a directory browser, the sub-tree structure is self-descriptive and easy to follow.
- **No unwanted inheritance of rights.** Users will not natively inherit rights to sub-OUs you don't want them to access. This eliminates the need to explicitly deny unwanted access rights that are being passed down the tree.
- **Robust network, Agent, and directory performance.** Typically, users who download large numbers of templates and policies generate more network traffic and a higher load on the directory than users who only download items relevant to their jobs. Grouping conserves your environment's resources and improves Agent response time.
- **Distributed administrative tasks.** Your templates are organized into easily controllable sets, and access rights determine who can manage which templates. You also have the ability to implement rights-based version control of your templates.
- **Low administrative overhead.** Controlling access at the template level requires setting permissions for each individual template via the Logon Manager Administrative Console; controlling access at the OU level is achieved via delegated administration using Microsoft and third-party management tools.

Figure 2 depicts a sample Logon Manager sub-tree whose design reflects the above best practices.

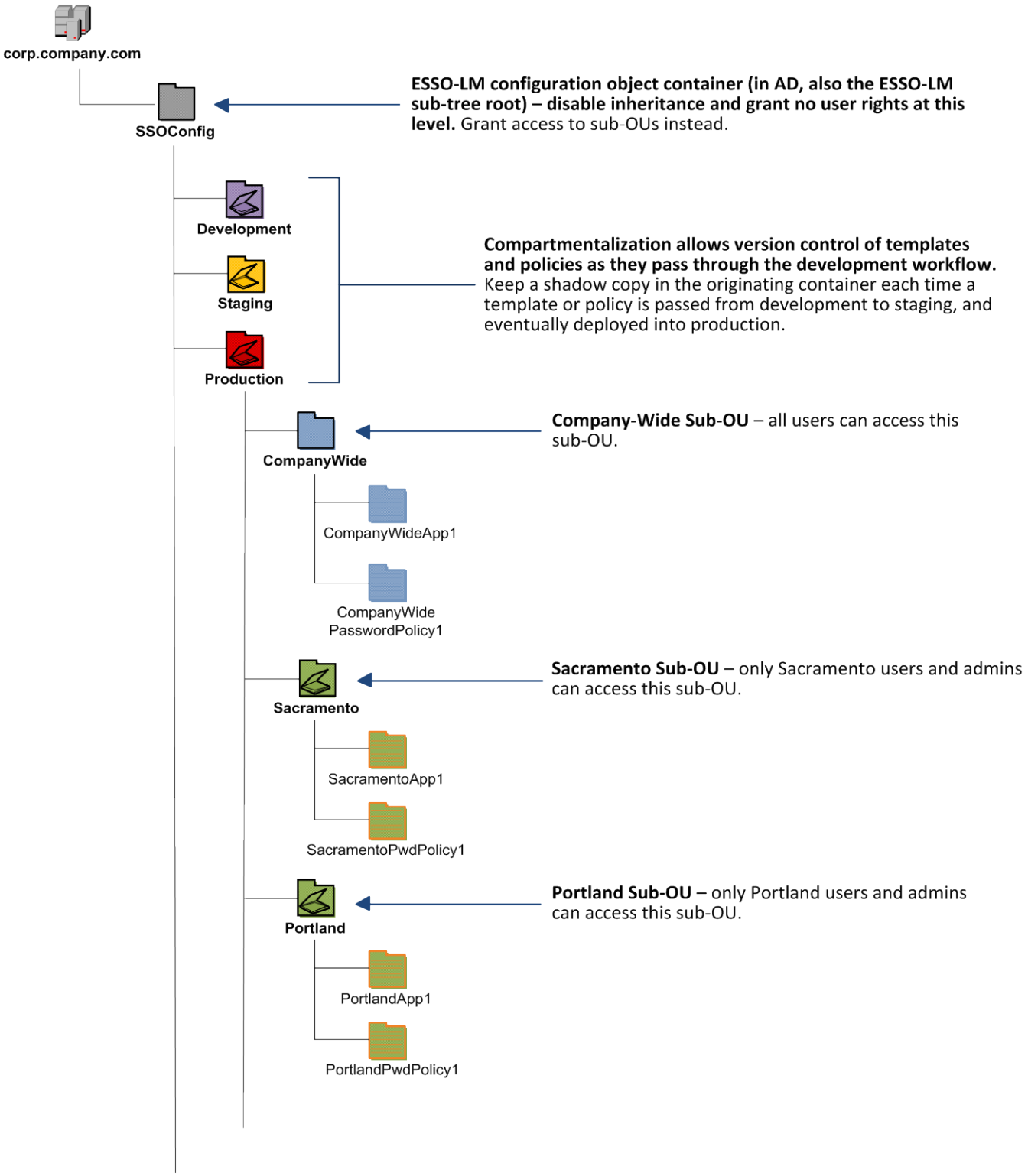


Figure 2 Recommended Logon Manager sub-tree design

In our sample scenario, users from the Portland division do not need access to applications used by the Sacramento division, and vice versa; therefore, each division's templates and policies live in dedicated sub-OUs under the root and one division cannot access another division's sub-OU. In the end, your environment will dictate the specifics of your implementation.

Note: Oracle highly recommends that you store templates and policies in individual OUs. To do this, you must enable the use of configuration objects.

If you are starting out with a flat model, but expect the number of users and provisioned applications to grow, create a sub-container under the root and use it to store your templates and policies as a flat file until you are ready to transition to a hierarchy. Monitor the performance of your environment as you add more users and provision more applications, and transition to a hierarchy sooner rather than later to minimize the required effort.

Achieving a Secure Configuration of Logon Manager

The behavior of the Logon Manager Agent, including its interaction with the directory, is governed by settings configured and deployed to the end-user machine by the Logon Manager administrator using the Administrative Console. The settings fall into one of the following categories:

- **Global Agent settings** are the "local policy" for the Agent**;** they are stored in the Windows registry on the end-user machine and are included in the Logon Manager MSI package to provide the Agent with an initial configuration during deployment. Global Agent settings are stored in `HKEY_LOCAL_MACHINE\Software\Passlogix` (32-bit systems) or `HKEY_LOCAL_MACHINE\Wow6432Node\Software\Passlogix` (64-bit systems).

  Caution: Users able to modify the HKLM hive can alter their global Agent settings and thus change the behavior of the Agent from the one originally intended. To ensure that a setting will not be changed by the end-user, deploy it through an **administrative override**.

- **Administrative overrides** take precedence over the global Agent settings stored in the Windows registry and constitute the "domain" policy for the Agent. Overrides are downloaded from the central repository by the Agent during synchronization and stored in the Agent's encrypted and tamper-proof local cache, which makes them immune to end-user alterations. When role/group security is enabled, administrative overrides can be applied on a per-user or per-group basis; they can also be applied enterprise-wide to enforce configuration consistency for all users.

  Note: Be conservative when planning your administrative overrides. Fewer overrides mean less data to store and transfer, and thus more efficient synchronization with the central repository. Reducing the number of overrides also simplifies troubleshooting by eliminating unknowns, as administrative overrides cannot be viewed on the end-user machine.

Global Agent settings together with administrative overrides constitute the *complete* configuration policy for the Agent. This guide describes settings recommended by Oracle to achieve a secure

deployment of Logon Manager and complements the information found in other Logon Manager documentation.

Warning: Settings such as domain names and user object paths should always be thoroughly tested before deployment and not deployed as administrative overrides unless absolutely necessary. A simple mistake, such as a mistyped domain name, can render end-user workstations unable to synchronize with the directory, in which case you will not be able to propagate a correction through the Administrative Console – changes will have to be made to user machines using other tools.

Recommended Global Agent Settings

This section lists the global Agent settings mandated by Oracle to achieve a secure Logon Manager configuration.

SSL Support

Logon Manager repository synchronizers ship with SSL support enabled and Oracle highly recommends that you do not disable it. Your environment should always utilize SSL for all connections to the Logon Manager and other repositories for maximum security.

Note: You must configure your domain controllers to use SSL before deploying Logon Manager. For instructions, see the following MSDN article: http://msdn.microsoft.com/en-us/library/aa364671(VS.85).aspx

Located in: Global Agent Settings → Live → Synchronization → [Synchronizer Name]

To re-enable (if disabled): Deselect the check box.

Use Configuration Objects (Active Directory Only)

On Active Directory deployments, Oracle highly recommends that you use directory objects for storing user and configuration data, allowing hierarchical storage, as well as role/group-based access control for individual containers, templates, and policies as described in Designing the Logon Manager Directory Sub-Tree. (If you disable this feature, Logon Manager will store all template and configuration data as a single flat file under the tree root.)

Located in: Global Agent Settings → Live → Synchronization

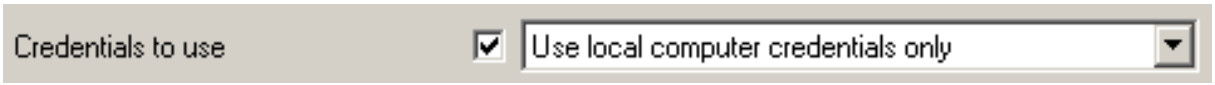
To enable: Select the check box, then select **Yes** from the drop-down list.

Select the Credentials to Use when Authenticating to the Directory

Use the **Credentials to use** option to select the credentials that Logon Manager should use when authenticating to the directory. Oracle recommends that you set this to **Use local computer credentials only** so that the user will not be prompted to reauthenticate if Logon Manager is unable to authenticate to the directory.

> **Note:** Do **not** leave this at the default setting, **Try local computer credentials; if it fails, use directory server account**. Doing so will cause an authentication failure (and the re-authentication prompt to appear, unless disabled) if the directory and the end-user machine are not part of the same domain.

Located in: Global Agent Settings → Live → Synchronization → [Synchronizer Name]



To set: Select the check box, then select the appropriate option from the drop-down list.

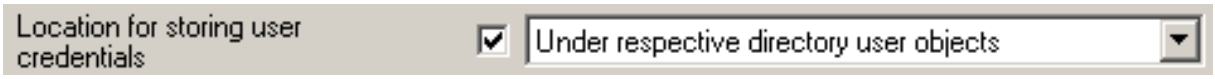
Store User Credentials Under Respective User Objects (Active Directory Only)

A major benefit of using Logon Manager with Active Directory is the ability to store user credentials under the respective user objects. Doing so simplifies administration as follows:

- Locating and viewing the credentials of individual users is quick and intuitive.
- Deleting a user from the directory automatically removes the user's application credential cache from under the respective user object.

> **Note:** This option will not work until you perform the necessary schema modification and permission assignment. For instructions, see the Logon Manager deployment guide for your repository.
>
> **Note:** When user credentials are stored under respective user objects and use of credential objects is enabled, you do not need to use the Locator object. (The Locator is a pointer object that tells Logon Manager where in the directory to look for templates, credentials, and other objects when using a flat directory model. For more information, see the Logon Manager deployment guide for your repository.

Located in: Global Agent Settings → Live → Synchronization → ADEXT



To enable: Select the check box, then select **Under respective directory user objects** from the drop-down list.

Recommended Administrative Overrides

This section lists the administrative overrides mandated by Oracle to achieve a secure Logon Manager configuration.

Store User Settings in a Secure Location (Active Directory Only)

Starting with version 11.1.2.0.0, when deployed on Microsoft Active Directory, Logon Manager configuration policies are now being stored in a secure, encrypted location in the repository. Oracle highly recommends that you migrate to this new settings storage schema by enabling the **Use secure location for storing user settings** option.

Warning: When upgrading from a previous version of Logon Manager, only deploy this override once all instances of Logon Manager have been upgraded to version 11.1.2.0.0; otherwise, once Logon Manager 11.1.2.0.0 synchronizes with the repository, all previous versions will no longer be able to synchronize with the repository for that user.

Located in: Global Agent Settings → Live → Synchronization → ADEXT

To set: Select the check box, then select **Yes** from the drop-down list.

Select the Primary Authenticator for End-Users

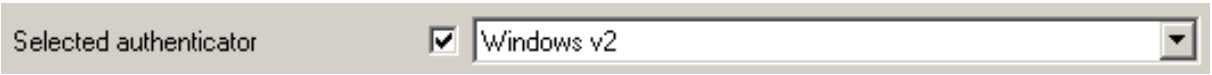
Oracle highly recommends that you select and configure the primary authenticator in the following scenarios:

- If you want to disable the FTU wizard, as described in the next section
- If you want users to authenticate only via the selected primary authenticator.

For information on configuring specific authenticators, see the Administrative Console help.

Note: If this setting is left blank and the FTU wizard is disabled, the first installed logon method (in descending alphabetical order) is automatically selected by default. To view the list of installed authenticators, temporarily enable the setting and examine its drop-down list.

Located in: Global Agent Settings → Live → User Experience → Setup Wizard



To set: Select the check box, then select the desired logon method from the drop-down list.

Warning: The original WinAuth (also known as WinAuth v1) authenticator has been deprecated and only remains in the product to enable upgrading from previous versions. **Do not** select it here unless explicitly instructed to do so by Oracle Support.

Use the Default Encryption Algorithm

Do not change the default encryption algorithm (**AES (MS CAPI)**) that Logon Manager uses to encrypt application credentials to retain compatibility with all supported operating systems. Not all algorithms supported by Logon Manager function with all operating systems.

Warning: All encryption algorithms except for AES (MS CAPI) have been deprecated and only remain in the product to enable upgrading from previous versions. **Do not** change the encryption algorithm unless explicitly instructed to do so by Oracle Support.

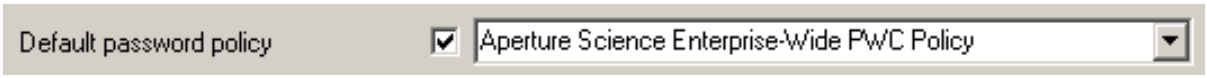
Located in: Global Agent Settings → Live → Security

To reset to default (if changed): Deselect the check box.

Create and Set the Company Password Change Policy

By default, Logon Manager ships with an inadequate default password change policy that must be replaced with a new policy which meets the security requirements of your organization. Include the name of your organization in the policy name to indicate that it is not a built-in policy. You must create this policy before setting this option; for instructions on creating a password change policy, see the Console help.

Located in: Global Agent Settings → Live → User Experience → Password Change



To set: Select the check box, then select the desired policy from the drop-down list.

Note: The policy set as the default password change policy is in effect enterprise-wide.

Force Reauthentication when Revealing Masked Fields

To prevent unauthorized access to stored application passwords, configure Logon Manager to prompt the user to authenticate when the "reveal masked fields" feature is invoked within the Agent. Configuring this policy as an administrative override will also prevent a rogue administrator from manually adding the setting to the local machine's registry and gaining unauthorized access to the local user's passwords if the setting is left unconfigured during initial deployment.

Located in: Global Agent Settings → Security

To set: Select the check box, then select **Yes** from the drop-down list.

Configuring Secondary Authentication for Logon Manager

In order to authenticate a user and grant access to stored credentials, Logon Manager offers a number of authentication methods implemented as authenticator plug-ins, with the most common method being a user name and password. On Active Directory environments, Logon Manager supports this authentication method through its Windows Authenticator v2 (WinAuth v2) plug-in.

When the user enrolls with Logon Manager for the first time, Logon Manager uses the user's unique authentication factors to secure access to their credential store. When deployed with WinAuth v2, Logon Manager, by default, will also prompt the user for a passphrase (unless explicitly configured otherwise) – this passphrase provides a means of secondary authentication to Logon Manager in case primary authentication fails.

Secondary Authentication via Interactive Passphrase Prompt

The interactive passphrase mechanism requires the user to provide an answer to a question presented during initial enrollment with Logon Manager and is the default and recommended secure secondary authentication method. (The question is defined by the administrator.) The user must supply the passphrase answer in order to authenticate to Logon Manager each time any of the factors used to encrypt the authenticator key have changed. The key advantages of this method are:

- **Acts as "second password" to the credential store.** The passphrase can be used to regain access to the credential store in the event the user's Windows password is no longer functional or accessible.
- **Prevents rogue administrator attacks.** A rogue administrator could potentially change a user's password, log on as that user, and gain access to the user's credential store; however, with a passphrase in place, the rogue administrator will not be able to gain access to the stored credentials without providing the passphrase answer.

Oracle highly recommends that your organization enforces the same cryptographic strength policy for passphrase answers as it does for passwords.

Secondary Authentication via Other Methods

WinAuth v2 also provides the following secondary authentication methods which you may choose to use instead of the passphrase method if your environment so requires:

- **Custom secondary authentication library** – WinAuth v2 provides a secondary authentication API which allows the passphrase answer to be programmatically supplied by an external secondary authentication library without the need for user interaction. While this API allows you to have complete control over the way the secondary key is delivered to Logon Manager during recovery, you are fully responsible for ensuring the security of the credential store key at all points during the recovery process.

  Note: For more information on the API, see the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*.

  - **User's Active Directory SID** – silently supplies the Active Directory SID of the currently logged on user as the passphrase answer to WinAuth v2. The encrypted SID is stored in the Agent's local cache. This secondary authentication method "follows" (roams with) the user on the network.
  - **Secure random key** – silently supplies a randomly generated key as the passphrase answer to WinAuth v2. This random key is stored in encrypted form within the user's Logon Manager credential store in Active Directory. Unauthorized access to the key is automatically restricted via Active Directory ACLs that are in effect for the user's credential store container.
  - **Windows Data Protection** – delegates secondary authentication to the Windows Data Protection service through the Windows Data Protection API (DPAPI). Since secondary access to the user's credential store is securely handled by DPAPI (locally to the user's workstation), it can be used as another "silent" secondary authentication method. However, since DPAPI's functionality is confined to the user's workstation, this secondary authentication method is not portable, as it will not "roam with" the user on the network and the user will not be able to use it if they log on to another workstation.

  Note: For detailed requirements and instructions on deploying Logon Manager with WinAuth v2, see the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*.

Understanding the GINA and Network Provider Components

The GINA (Graphical Identification aNd Authentication) library and Network Provider service components of WinAuth v2 provide integration with the user authentication mechanism in the Windows operating systems. The GINA component hooks into the GINA chain on Windows XP and Windows Server 2003/2003 R2 systems, while the Network Provider service allows integration with and Windows 7 and Windows Server 2008/2008 R2, which do not use a GINA library. The Network Provider service also enables integration on Windows XP systems on which changes to the GINA library are not permitted or feasible. This integration provides the following advantages:

- **Eliminates the need for double authentication.** Without this integration, the user would need to provide their Windows credentials twice – once to the operating system in order to log on,

unlock the desktop, or exit a secure screensaver, and again to Logon Manager in order to access stored credentials.

- **Unlocking of the credential store is transparent to the user.** Logon Manager automatically intercepts the user's Windows credentials during logon and unlocks the credential store so that the user does not need to authenticate to Logon Manager in order to use automatic single sign-on, unless Logon Manager has been configured otherwise.

Note: If you are deploying Logon Manager with WinAuth v2 in a Password Reset environment in which client machines run either Windows XP or Windows Server 2003/2003 R2, do not install the Network Provider component, as it is incompatible with Password Reset; only install the GINA library.

Ensuring Secure Response to Applications

Once Logon Manager successfully detects an application's logon or password change form, it will respond by injecting and submitting credentials to the application. Depending on the design of the target application, Logon Manager can use one of the following methods to interact with the fields and controls in the target form.

Control IDs

This is the default and preferred form interaction method for most applications. This method uses the Windows API (Windows applications), the appropriate browser helper object (Web applications) or HLLAPI (mainframe applications) to identify and interact with the fields and controls within the form. In a Windows API-compliant application, each field and control exposes a unique Control ID to the operating system. The Agent detects these Control IDs and binds to them specific sign-on functions that signify the purpose of the object represented by the Control ID, such as the password field or the "submit" button.

Note: If some or all of the Control IDs exposed by the application are non-unique or dynamic, the Agent can substitute ordinals – sequential ID numbers assigned to each item in the window from top to bottom, left to right – to uniquely identify the detected fields and controls.

If a field or control does not expose its Control ID, or if there are additional actions required to complete the sign-on event, such as selecting a check box or manually setting focus on a specific field, you may also have to use the "SendKeys" method, in tandem with Control IDs, to interact with the target form.

"SendKeys"

This method allows the Agent to interact with the target application by emulating user input, such as keystrokes and mouse clicks. Use this method if the Agent is unable to programmatically detect or interact with the target fields and controls. For example, if an application does not expose Control IDs for any of its fields and controls, you will need to send individual keystrokes to populate the fields,

mouse clicks or **Tab** keystrokes to toggle between fields, and a final mouse click to engage the **Logon** button.

> **Note:** If you configure a form definition to use the "Control IDs" method and Logon Manager fails to inject credentials programmatically due to application incompatibility, by default Logon Manager will automatically fall back to the "SendKeys" method and retry injection. You can disable this fallback behavior by as described in the *Logon Manager Template Configuration and Diagnostics* guide for your target application type.

Control IDs with "SendKeys"

This method is a "best-of-both-worlds" combination of the two above methods and is the preferred way of solving sign-on scenarios that require actions that cannot be performed programmatically. Control IDs are used to interact with the form wherever possible, while keystrokes and mouse clicks are sent to accomplish tasks such as setting field focus, selecting a check box, and other actions that the Agent cannot perform programmatically within the target application.

> **Note:** To achieve this "mixed" mode, configure the form to initially use the "SendKeys" response method, then configure the desired "SendKeys" actions; while configuring the actions, enable the **Inject directly into control** option for actions that you wish to retain the "Control IDs" programmatic response method.

For example, if fields must be populated in a specific order due to cascading validation (i.e., the password field becomes active only once the user name field has been populated), you would use Control IDs to inject credentials into the fields, but send a **Tab** keystroke via "SendKeys" between each field injection to advance field focus.

Which Method to Use?

Because the "SendKeys" method emulates keystrokes, it is theoretically possible, with the right timing, for a user to switch focus to another application while Logon Manager is injecting credentials and thus capture them in plain text. For this reason, Oracle recommends using the Control ID injection method whenever possible and only falling back to the "SendKeys" method when programmatic injection is not feasible.

Configuring User Access at the Template Level

When configuring an application template, you can use its Security tab to limit which users and groups have access to single sign-on functionality for the target application. Oracle recommends assigning specific users to groups and then limiting access to applications via user groups rather than individual users to ensure a more robust administration model, unless your environment specifically requires per-user access control granularity.

Securing Password Reset

Password Reset consists of several client-side and server-side components that communicate with one another via SSL-encrypted HTTP and access a data repository over an SSL-encrypted channel.

Securing Password Reset on the Client Side

On the client side, Password Reset hooks into the Windows logon mechanism using either a GINA library stub (Windows XP) or a credential provider and a system service running under the `LocalSystem` account. Either mechanism allows Password Reset to add password reset functionality to the standard Windows logon dialog, either by adding a banner or a hyperlink that launches a locked-down Internet Explorer window that connects (via HTTP with SSL) to the server-side Password Reset Web applications described below. Assuming that the server-side components are configured for SSL connectivity, the client-side configuration is secure by default and does not require additional hardening.

> **Note:** After configuring the server-side components to use SSL, make sure that the Web application URLs on end-user machines are updated to use the HTTPS protocol.

Securing Password Reset on the Server Side

On the server side, Password Reset runs IIS-hosted Web applications as well as a Windows system service that together provide the password reset, challenge question quiz, and administration functionality, as well as user interfaces for each. They also provide the challenge question functionality to Universal Authentication Manager.

The Web applications are `EnrollmentClient`, `ResetClient`, `ManagementClient`, and `WebServices`. The `ResetClient` and `WebServices` Web applications require that a limited-privilege domain user account (`SSPRWeb`) is created and assigned as the sole account able to access the pages within them as well as modify user data within the repository.

The `EnrollmentClient` and `ManagementClient` applications, as well as the `Administration.aspx` page of the `WebServices` application are configured for access by the domain user account currently logged on to the Password Reset-enabled end-user workstation. Configuration steps are described in the *Enterprise Single Sign-On Suite Plus Installation Guide.*

Oracle strongly recommends that you configure the Password Reset Web applications within IIS to use SSL. To enable SSL support for Password Reset, you must create and install an X.509 SSL certificate for the IIS Web sites serving the Password Reset Web applications. (The certificate is issued by a Certificate Authority (CA), which can be a commercial entity or a software application on the target local machine.) You must then update your end-user workstations with secure (HTTPS) URLs to the Password Reset Web applications. Instructions are provided in the *Enterprise Single Sign-On Suite Administrator's Guide*.

> **Caution:** Oracle highly recommends that you do not disable SSL functionality to maintain maximum security.

Password Reset also utilizes a Windows system service, `SSPRChangePasswordSvc.exe`, which runs in the background and is responsible for the actual changing of each user's password once the user has passed the Password Reset challenge quiz. This service requires a limited-privilege domain account (`SSPRReset`) that possesses only the permissions required to change user account passwords as well as write the lockoutTime and pwdLastSet values in Active Directory. The configuration steps and exact permissions that must be assigned to this account are described in the *Enterprise Single Sign-On Suite Plus Installation Guide.*

Password Reset stores user data in a supported repository: Active Directory, ADAM/AD-LDS, LDAP directories such as Oracle Internet Directory and Oracle Virtual Directory, as well as Oracle and Microsoft SQL databases. During installation, the installer creates an organizational unit (directory-based repositories) or databases and tables (database-based repositories) and grants the `SSPRWeb` domain account full access to the newly created container or database and all its contents. No other account is given any kind of access to the Password Reset container or database unless the administrator explicitly grants such access through their own choice.

Securing Kiosk Manager

Kiosk Manager allows multiple users to use a single workstation in a kiosk environment, such as a medical office or a hospital, by allowing one or more “sub-sessions” within the context of a single Windows account session.

Oracle recommends that you utilize one or more of Kiosk Manager’s session security features:

- Lock the session when the user hits Ctrl-Alt-Del,
- Lock the session when the screen saver engages,
- Disable Task Manager, the Run command, as well as Start Menu and Windows taskbar access using the mouse and Windows hotkeys when session is locked so that applications cannot be launched, switched, or terminated by the user,
- Stay on top of all other windows and prevent other applications from stealing focus.

Provided that Logon Manager has been securely deployed and configured as described in Securing Logon Manager, no extra work is necessary to secure Kiosk Manager. This is because the Kiosk Manager plug-in within Logon Manager uses Logons Manager’s synchronization mechanism to interact with the repository, eliminating the need for a dedicated connection. Connection and data security is ensured by Logon Manager’s built in encryption mechanisms, provided the repository connection is utilizing SSL.

To prevent a user from accessing the applications of another user within another KM session, you should follow industry standard best practices for securing a public end-user workstation. Specifically, the Windows account under which Kiosk Manager is to run should be stripped from all privileges except those that permit the launching and use of the required target applications so that the user cannot terminate Kiosk Manager or other users’ applications.

You should also always set an inactivity timer which will lock the user’s session after a short period of inactivity – for example, when the user walks away from the kiosk to tend to a patient.

Securing Provisioning Gateway

Provisioning Gateway allows administrators to remotely provision application credentials to Logon Manager users using either the included Provisioning Gateway Web console or by interfacing with Oracle and third-party identity management solutions.

On the server side, Provisioning Gateway runs as two Web applications hosted via Microsoft IIS:

- **Provisioning Gateway Web Service** – enables the remote provisioning functionality, including receiving provisioning instructions from the Web Console service or external identity management solutions (via appropriate connector plug-ins) and writing them to the Logon Manager repository.
- **Provisioning Gateway Web Console** – provides a management front-end to the Provisioning Gateway Web Service, enabling configuration of Provisioning Gateway and credential provisioning.

On the end-user side, a plug-in within Logon Manager reads the provisioning instructions stored in the Logon Manager repository during each synchronization event and executes them by adding, modifying, or deleting application credentials from the user's Logon Manager credential store.

Securing Provisioning Gateway on the Client Side

Provided that Logon Manager has been securely deployed and configured as described in Securing Logon Manager, no extra work is necessary to secure Provisioning Gateway on the client side. This is because the Provisioning Gateway plug-in within Logon Manager uses Logons Manager's synchronization mechanism to interact with the repository, eliminating the need for a dedicated connection. Connection and data security is ensured by Logon Manager's built in encryption mechanisms, provided the repository connection is utilizing SSL.

Securing Provisioning Gateway on the Server Side

To secure Provisioning Gateway on the server side, you must do the following:

- Make sure you have structured and configured your Logon Manager repository in a secure manner as described in Securing Logon Manager.
- Configure the Provisioning Gateway Web services listed in the previous section to only accept SSL connections.
- Configure the Provisioning Gateway Web Console service to use an "https" URL to connect to the Provisioning Gateway Web service.
- Create a dedicated account within the domain hosting the Logon Manager repository that the Provisioning Gateway Web service will use to connect to the repository, and limit that account's access privileges to the bare minimum required for Provisioning Gateway to function properly. Create the account as follows:

- The service account must be a member of the Domain Users group within the domain to which Provisioning Gateway servers belong.
- The service account must be a member of the local Administrators group on each Provisioning Gateway server machine.
- The default name for this account is PMSERVICE; however, you may name the account as required by your environment and configure Provisioning Gateway to use the customized name.

The configuration instructions are provided in the *Enterprise Single Sign-On Suite Administrator's Guide*.

Securing Universal Authentication Manager

Logon Methods

Note: Universal Authentication Manager can be deployed in enterprise (centrally-managed, repository-based) or local (standalone) mode. The recommendations in this section are meant for enterprise-wide enforcement via administrator-configured policies and thus apply mostly to Universal Authentication Manager's enterprise mode only. For more information, see the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*.

Universal Authentication Manager supports fingerprint, proximity/smart cards, a challenge questions quiz, and the Windows password as logon methods. Follow the guidelines below to maximize security when using each method:

- **Use a high-strength PIN (fingerprint, proximity/smart cards only).**
  - Require a PIN in conjunction with your chosen logon method to enforce two-factor authentication. Unless your environment specifically disallows the use of a PIN, it should be configured and required at all times.
  - Enforce high PIN complexity by configuring each logon method's PIN policy for increased minimum length, requiring multiple character types (uppercase, numeric, extended), and so on.
  - When using smart cards, configure Universal Authentication Manager to use the smart card's built-in PIN and reconfigure the PIN policy stored on the card to increase PIN complexity as described above.
- **When using smart cards, configure the cards to use your enterprise's Public Key Infrastructure (PKI) and install and configure a certificate revocation plug-in.** (For more information on certificate revocation plug-ins, contact Oracle Support.)
- **Enforce strong authentication as the only permitted logon methods.** Disallowing the Windows password as a logon method to Universal Authentication Manager decreases the chance of brute-force and social engineering attacks.
- **If your environment requires passwords, enforce an enterprise-wide complex password policy.** Your password policy should enforce highly complex passwords to minimize the chance of a successful brute-force attack.
- **Enforce unique challenge questions.** Create and enforce unique challenge questions to which answers cannot be easily guessed.

Repository Connection

Note: Universal Authentication Manager can be deployed in enterprise (centrally-managed, repository-based) or local (standalone) mode. For more information, see the *Oracle Enterprise Single Sign-On Suite Administrator's Guide*.

Universal Authentication Manager securely stores user authentication and policy data within an Active Directory based-repository. Data stored in and transmitted between Universal Authentication Manager and the repository is always encrypted and thus not decipherable by a rogue administrator viewing the repository content directly. For added security, Oracle also recommends configuring your repository for SSL connectivity to further increase security.

Service Account (Enterprise Mode Only)

When running Universal Authentication Manager in enterprise (repository synchronization) mode, you must create and configure a domain account that will allow Universal Authentication Manager to connect to and make changes in its repository. For maximum security, you must:

- Strictly follow the repository configuration instructions, including the minimum necessary privilege assignment, described in the *Enterprise Single Sign-On Universal Authentication Manager Administrator's Guide*. Do not assign any additional privileges to the service account.
- Configure the Universal Authentication Manager repository containers to be accessible only by this service account and no other user.

Note: This account must also be granted the "Run as a Service" privilege locally on the end-user workstation in order to allow Universal Authentication Manager to function.

User Policies (Enterprise Mode Only)

When deploying Universal Authentication Manager in enterprise mode, Oracle highly recommends that you do not rely on configuration defaults and instead deploy enterprise-wide policies that explicitly enforce each Universal Authentication Manager setting so that users cannot change them. When an explicit policy is in effect, Universal Authentication Manager settings cannot be modified by the end-user.

Synchronization with Password Reset

If you are deploying Universal Authentication Manager with the Challenge Questions logon method and wish to use Password Reset to centrally configure the challenge questions and store the user's enrollment data, Oracle recommends that you set up your Password Reset installation to only accept SSL connections for maximum security. For more information on integrating with Password Reset, see the *Oracle Enterprise Single Sign-On Universal Authentication Manager Administrator's Guide*.

Securing the Reporting Service

The Reporting Service accepts events occurring within each Suite application and stores them in the database configured by the Suite administrator. There are two communication pathways through which the Reporting Service receives and transmits event information:

- **Suite Application and Reporting Service** – the event data is sent to the Reporting Service securely and automatically by each running Suite application. Before transmitting the data to the service, each Suite application verifies the service's digital signature and will not transmit the data if the verification fails. Additionally, before accepting event data from any application, the service verifies the application's digital signature and will not accept event data if the verification fails. Thus, a bi-directional trust must be established between the transmitting application and the service itself for data exchange to occur. Programmatically impersonating either a Suite application or the Reporting Service is thus impossible.
- **Reporting Service and database** – the Reporting Service is capable of opening a secure connection to the database holding its event data repository; however, you are responsible for configuring your database for secure communication and providing the Reporting Service with the correct connection string that enables secure communication. For example:

  ```
  "Provider=SQLOLEDB;Data Source=myServerName;Initial
  Catalog=myDatabaseName;Integrated Security=SSPI;Use Encryption
  for Data=True"
  ```

  In the above example, the `Integrated Security=SSPI` and `Use Encryption for Data=True` setting ensure a secure connection to the database. For more information, consult your database system's documentation. For steps on configuring a database instance for the reporting service, see the *Enterprise Single Sign-On Suite Plus Administrator's Guide*.

Securing Anywhere

For maximum security, Oracle recommends that you generate an SSL certificate that will be used to sign custom installation packages created with Anywhere. For instructions on creating and exporting such a certificate, see the *Oracle Enterprise Single Sign-On Suite Administrator's Guide.*