Agile Product Lifecycle Management

Agile PLM Security Guide

Version 9.3.2

ORACLE

E28772-01

December 2012

CONTENTS

Preface

The Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Note   To read the PDF files, you must use the free Adobe Acrobat Reader version 7.0 or later. This program can be downloaded from the Adobe Web site http://www.adobe.com.

The Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html can be accessed through **Help > Manuals** in both Agile Web Client and Agile Java Client. If you need additional assistance or information, please contact My Oracle Support (https://support.oracle.com) for assistance.

Note   Before calling Oracle Support about a problem with an Agile PLM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Readme

Any last-minute information about Agile PLM can be found in the Readme file on the Oracle Technology Network (OTN) Web site http://www.oracle.com/technetwork/documentation/agile-085940.html

Agile Training Aids

Go to the Oracle University Web page http://www.oracle.com/education/chooser/selectcountry_new.html for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Document Scope

This chapter includes the following:

This document provides IT users and Agile PLM administrators with the information needed securely set up and deploy Agile PLM.

Documentation Audience

This document is written for IT users and Agile PLM administrators who will be setting up Agile PLM. It is assumed that those reading this documentation have a solid understanding of security concepts. The audience should also have basic knowledge of roles and privileges in Agile PLM.

Guide to this Document

This guide provides information needed to help you to securely set up and configure Agile PLM.

The guide is organized as follows:

▫ Overview of Agile PLM on page 3 gives an overview of Agile PLM and its modules.

▫ Overview of Security Fundamentals on page 5 provides an overview of basic security principles which should be considered while setting up Agile PLM.

▫ How to Perform a Secure Agile PLM Installation on page 7 provides guidance on how to securely install the Oracle Database Server, Oracle WebLogic Server, Agile PLM, and the Agile PLM Database.

▫ How to Securely Configure Agile PLM on page 13 provides information on how to use Agile PLM's security features to securely configure your deployment. User authentication and authorization is discussed in this chapter. Additionally, application-level configuration properties used to secure the application are discussed here

▫ Security Considerations for Developers on page 17 provides information needed for developers to extend the Agile PLM application or produce applications using Agile PLM as a platform.

Agile PLM Overview

The Agile PLM suite of solutions covers five primary areas of product lifecycle management:

▫ **Agile Product Collaboration** (PC) — Management and collaboration of product record information throughout the product lifecycle, across internal organizations and the extended supply chain. Accessed through Web Client and Java Client.

▫ **Agile Product Governance & Compliance** (PG&C) — Management and tracking of all substances and materials contained by any item or manufacturer part, allowing companies to meet substance restrictions and reporting requirements, design recyclable products, minimize compliance costs, and eliminate noncompliance on future products. Accessed through Web Client.

▫ **Agile Product Portfolio Management** (PPM, formerly Program Execution) — Integration of program and product information, streamlining business processes across the product lifecycle and across a portfolio of programs. Accessed through Web Client.

▫ **Agile Product Quality Management** (PQM, formerly Product Service & Improvement) — Integration of customer, product, quality, and regulatory information with a closed-loop corrective action system. Accessed through Web Client and Java Client.

▫ **Agile Product Cost Management** (PCM) — Management of product costs across the product lifecycle and synchronization of product cost data and processes. Accessed through Web Client.

▫ **Agile Recipe & Material Workspace** (RMW) — Management of biotechnological and pharmaceutical products, as well as improvement of business productivity, visibility, scientific outcomes, and proactive compliance during the product development lifecycle. Accessed through Web Client. For more information, see *Agile PLM Getting Started with Recipe & Material Workspace*.

Agile administrators use Agile Java Client to set up and maintain settings for these solutions.

The Agile Application Server, the foundation of the Agile suite, manages data stored in the Agile database. All Agile data is contained or organized in business objects that are set up by the administrator, and specified and used by the enterprise's Agile users. For instance, the administrator configures the Parts class of objects, and users create and deploy specific instances of the kinds of Parts made available to them. Business objects is a general term that implies objects created from the classes available to the enterprise, but other entities in Agile are also objects, such as workflows, searches, reports, and so forth.

The following figure shows relationships between the Agile functional components, the primary client applications used to manipulate the data (Agile Web Client and Java Client), and the Agile Application and Database Servers (the database where the data is stored).

AGILE PLM FUNCTION

Administrative Settings

Product Collaboration

Product Quality Management

Product Cost Management

Product Portfolio Management

Product Governance & Compliance

Recipe & Material Workspace

CLIENT APPLICATION USED FOR WORKING WITH DATA

Agile Java Client

Agile Web Client

DATA STORAGE

Application & Database Servers (hold the Agile database)
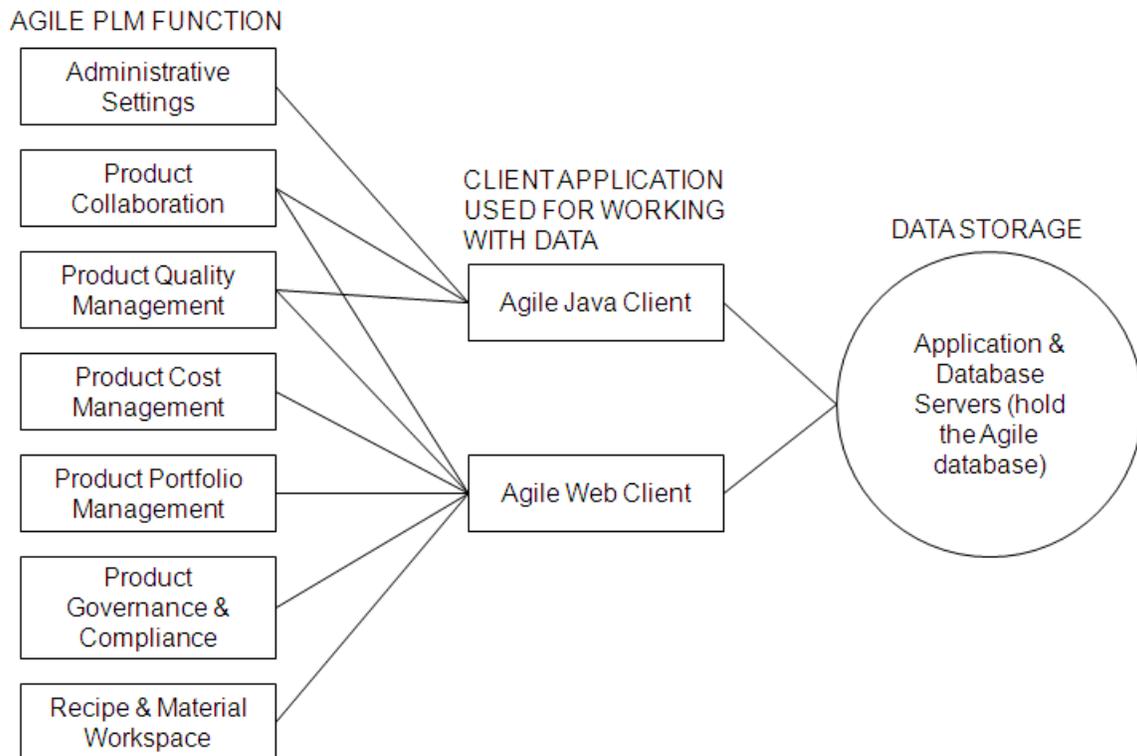
Overview of Security Fundamentals

This chapter includes the following:

This chapter provides an overview of general security considerations and recommendations.

Basic Security Considerations

The following principles are fundamental to using any application securely.

Keep Software Up-To-Date

One principle for good security practice is to keep all software versions and patches up-to-date. To ensure that you have the most current and updated Agile PLM software for the latest version, regularly check the Oracle Critical Patch updates page.

Restrict Network Access to Critical Services

Keep both the Agile PLM application and the database behind a firewall. In addition, place a firewall between the middle-tier and the database. The firewall provides assurance that access to these systems is restricted to a known network route, which can be monitored and restricted, if necessary. As an alternative, a firewall router substitutes for multiple, independent firewalls.

If you cannot use firewalls, then configure the TNS Listener Valid Node Checking feature (it restricts access based upon IP address). Restricting database access by IP address often causes application client/server programs to fail for DHCP clients.

To solve this problem, use any of the following:

- static IP addresses
- software VPN
- hardware VPN
- software VPN and hardware VPN
- Windows Terminal Services or its equivalent

Follow the Principle of Least Privilege

The principle of least privilege states that users should be given the least amount of privilege to perform their jobs. Over-ambitious granting of responsibilities, roles, grants, etc., especially early on in an organization's life cycle when people are few and work needs to be done quickly, often leaves a system wide open for abuse. User privileges should be reviewed periodically to determine relevance to current job responsibilities.

Monitor System Activity

System security stands on three legs: good security protocols, proper system configuration and system monitoring. Auditing and reviewing audit records address this third requirement. Each component within a system has some degree of monitoring capability. Follow audit advice in this document and regularly monitor audit records.

Keep Up To Date on Latest Security Information

Oracle continually improves its software and documentation. Check this note yearly for revisions.

How to Perform a Secure Agile PLM Installation

This chapter includes the following:

This chapter describes a recommended deployment topology for your PLM system and then provides recommendations on how to securely install and configure the Agile PLM system.

Understanding the Agile PLM Environment

When planning for a secure Agile PLM implementation, consider the following:

□ **Which resources need to be protected?**

- You need to protect customer data, such as part numbers, file attachments, etc.

- You need to protect internal data, such as proprietary source code.

- You need to protect information in databases accessed by the Agile PLM server and the availability, performance, applications, and the integrity of the Web site.

- You need to protect system components from being disabled by external attacks or intentional system overloads.

□ **Who are you protecting data from?**

For example, you need to protect your subscribers' data from other subscribers, but someone in your organization might need to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, it is possible that a system administrator can manage your system components without needing to access the system data.

□ **What will happen if protections on a strategic resources fail?**

In some cases, a fault in your security scheme is nothing more than an inconvenience. In other cases, a fault might cause great damage to you or your customers. Understanding the security ramifications of each resource will help you protect it properly.

Recommended Deployment Topologies

The following figure shows the general topology that is recommended for a secure Agile PLM installation.
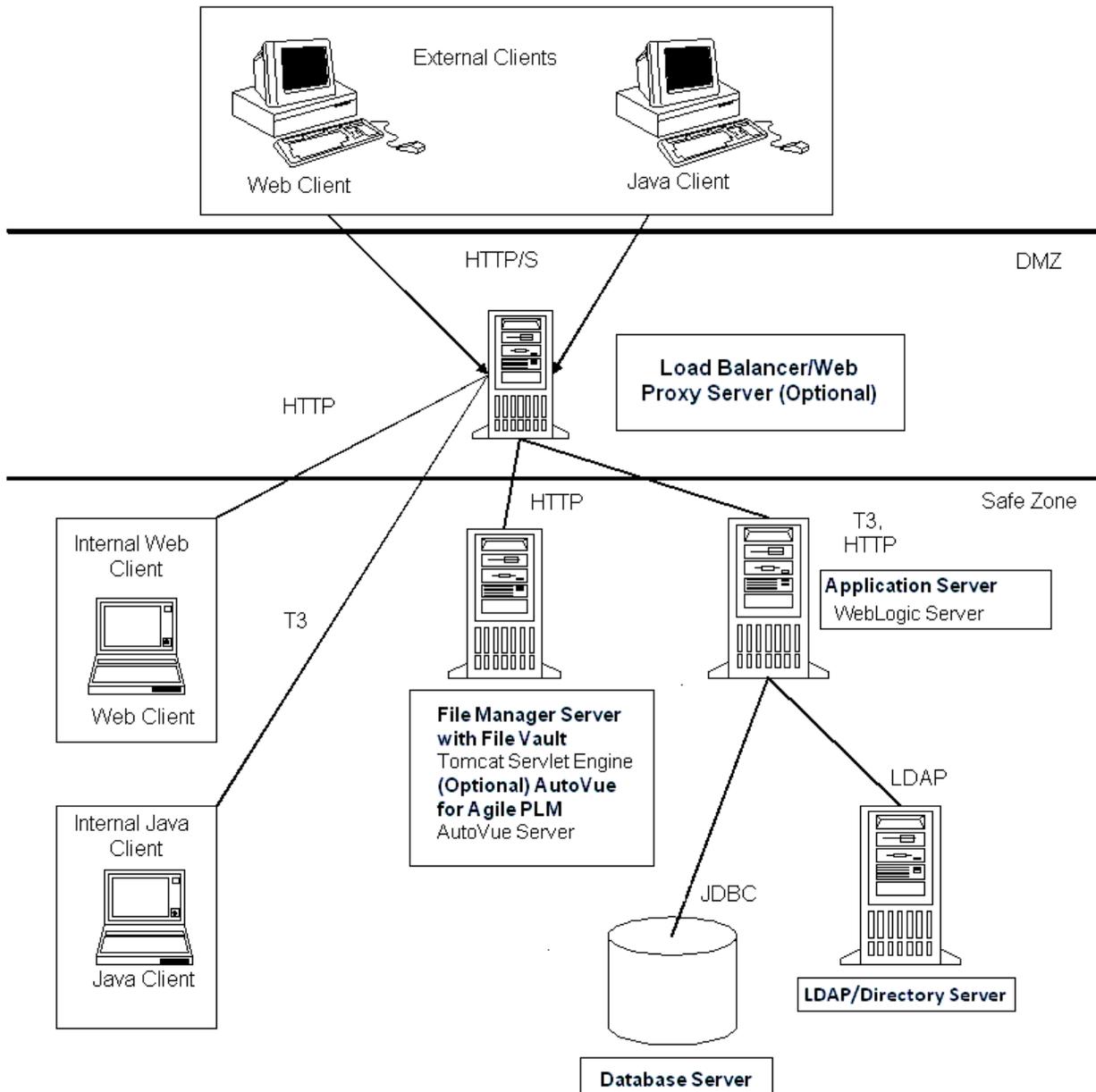
The components included in the topology diagram are defined below:

▫ **Agile PLM Web and Java Clients** - Agile PLM includes two clients, a Web client and a Java client. The Web client is a thin HTML client that uses firewall-friendly protocols (HTTP/S). The Java client is a Java-based client that can use application server-specific protocols, such as T3 for Oracle WebLogic, to connect to the server.

▫ **(optional) Proxy** - The hardware load balancer/proxy brokers client communications without compromising the security of your internal network. Clients communicate through the load balancer with the application server. There are no Agile software components running on the hardware load balancer. They are usually deployed in the Demilitarized Zone (DMZ) where it proxies requests from outside the corporate firewall to the application server in the Safe Zone.

   We recommend communication using HTTP over SSL (HTTPS) for the most secure deployment.

   • For standalone application server deployments, both the load-balancer and web server components are optional.

   • For deployments where the application server is clustered/redundant, a load-balancer is required and the web server is optional.

   Refer to the documentation for your proxy server to determine the most secure configurations.

▫ **Agile PLM Application Server** - The Agile Application Server is the center of the Agile system, the base for the PLM platform, where all common services and business logic reside for the entire solution. The Agile Application Server runs on industry-leading J2EE application servers. As the System Configuration Overview figure illustrates, all client servers and users connect to the Application Server either directly or indirectly. The application server connects to the components in a persistence layer where product content is stored.

   We recommend communication using HTTP over SSL (HTTPS) for the most secure deployment.

▫ **Agile PLM Database Server** - The Agile Database Server persists or stores all product content and system settings. Agile's database server runs on Oracle 11g or 12c.

▫ **(optional) LDAP / Directory Server** - In an effort to better support the industry standard authentication schemes, Agile PLM supports Lightweight Directory Access Protocol (LDAP) based authentication. LDAP support enables you to integrate Agile with existing directory servers so user accounts can be managed in one place. Integrating with LDAP is optional. Users can be managed within Agile without a directory server. There are no Agile software components deployed on the Directory Server.

   If using LDAP, we recommend communication using LDAPS for the most secure deployment.

▫ **PLM File Manager / AutoVue Server** - The Agile PLM File Manager component provides file upload/download functionality for the Agile PLM application. We recommend communication using HTTP over SSL (HTTPS) for the most secure deployment. The AutoVue Server component provides file viewing functionality for the Agile PLM application.

▫ **PLM File Vault** - The Agile PLM File Vault is comprised of one or more file system(s) on which the Agile PLM File Manager component stores and retrieves files uploaded/downloaded in the Agile PLM application.

Note	Oracle suggests that you create a similar Network Diagram to illustrate your deployment's specific network topology, including servers, routers, firewalls, etc. This document may be requested by Oracle Support should a network connectivity issue arise.

Installation - Prerequisites

Before installing Agile PLM, you must install and configure Oracle Database Server and Oracle WebLogic Server. The following sections include recommendations on how to set these products up to ensure a secure configuration.

Installing the Oracle Database Server

For the latest information on installing Oracle Database Server in a secure manner, refer to the *Oracle Database Security Guide* and make necessary configuration changes. For additional information, refer to the "Installing Oracle Database Server" chapter in the *Agile Product Lifecycle Management Database Installation Guide*.

Installing Oracle WebLogic Server

After installing the Oracle Database Server you should install the Oracle WebLogic Server.

For the latest information on how to install WebLogic Server, refer to the appropriate Oracle WebLogic Server documentation.

Additionally, Oracle recommends that you:

□ Deploy WebLogic Server using SSL.

□ After installation, change the WebLogic administrator username and password.

□ Secure WebLogic Server by placing it behind a proxy server.

The following WebLogic Server documents contain information that is relevant to the WebLogic Security Service:

□ *Understanding Security for Oracle WebLogic Server -* Summarizes the features of the WebLogic Security Service, including an overview of its architecture and capabilities. It is the starting point for understanding WebLogic security.

□ *Securing Oracle WebLogic Server -* Explains how to configure security for WebLogic Server and how to use Compatibility security.

□ *Securing a Production Environment for Oracle WebLogic Server -* This document highlights essential security measures for you to consider before you deploy WebLogic Server into a production environment.

Installing Agile PLM and Database

This section describes best practices to be followed while using the Agile PLM and database installers.

For the latest information on installing Agile PLM, including the supported operating systems, refer to the *Installing Agile PLM for WebLogic* guide. The following users are created out-of-box for the application to start up correctly and function as expected: admin, agileuser, etluser, ifsuser, propogation, superadmin.

Note	These OOB users should not be dropped or modified without consulting Oracle Support, as this will affect the functionality of the product.

For the latest information on installing the Agile PLM database schema, refer to the *Agile Database Installation Guide.*

Additionally, Oracle recommends that you:

□ Use strong passwords.

□ Deploy with SSL.

□ Use the Agile PLM system for authentication.

□ Use Oracle Platform Components such as OID or OAM for authentication requirements.

Optional Component Configuration

To ensure a secure configuration, consider the following recommendations for optional components.

Configuring AutoVue (Optional)

Refer to the *AutoVue Security Guide* for information about configuring AutoVue securely.

How to Securely Configure Agile PLM

This chapter includes the following:

This chapter describes how Agile PLM uses the following security features to provide data protection:

- **Authentication** - allows only permitted individuals to get access to the system and data.

- **Access Control (Authorization)** - provides authorized individuals access control to system privileges and data.

- **Audit** - allows Administrators to detect attempted breaches of authorization and attempted (or successful) breaches of access control.

Password Policy

A password policy is a set of rules dictating how to use passwords. Some of the rules a password policy sets are:

- The maximum length of time a password is valid

- The minimum number of characters in a password

Password policies play an important role when attempting to access a directory. The directory server ensures that the password entered adheres to the password policy.

Configuring and Using Authentication

Agile PLM supports the Lightweight Directory Access Protocol (LDAP), Single Sign-On (SSO), and database authentication configurations.

The three supported authentication configurations are discussed below.

LDAP-based Authentication

LDAP is an application protocol for querying and modifying directory services running over TCP/IP.

Agile PLM supports LDAP authentication through the Agile Directory Server Integration Module. You can integrate Agile with your existing directory server to manage your users in one place. This approach can be fully integrated into Agile PLM, for these supported directory servers:

- Oracle Internet Directory Server

- Microsoft Active Directory Server

□   Sun Java System Directory Server

If you chose to manage your user accounts through a directory server (instead of the database) during installation, then all new users are added, and certain user attributes are configured, only through the directory server. Users need to be synced from the LDAP system to the Agile PLM database.

For more information, refer to the "LDAP" chapter in the *Agile PLM Administrator Guide*.

SSO-based Authentication

Agile PLM has the possibility of integrating aspects of your PLM system with Single Sign-On (SSO) capability. SSO is a Web-based solution that can be enabled only for Agile Web Client.

Single Sign-On integrates with the centralized security management system, other business and training applications, and improves user productivity in the Agile Web Client environment. With SSO configured and enabled for your PLM system, a user that has signed in to the system once (for instance, through the corporate portal) is not prompted again by a "login" dialog.

Agile PLM is certified on the following Single Sign-On solutions:

□   Oracle Access Manager (OAM)

□   NT LAN Manager (NTLM)

Note the following:

□   Agile SDK code cannot connect to an Agile application URL protected by SSO.

□   Users cannot develop Java Web Service client code and connect to an Agile Web Service protected by SSO.

□   Webdav (AgileDrive) cannot connect to an Agile Application Server URL protected by SSO.

□   Web Service clients or SDK code must connect directly to Agile server nodes with actual WebLogic ports or set up an alternate proxy that is not protected by SSO.

For more information, refer to the "Configuring Single Sign-On" chapter in the Agile PLM Administrator Guide. The chapter also includes a helpful diagram of the Agile SSO Plug-in Architecture.

URL PX-based SSO

Customers use Process Extensions (PX) to extend Agile UI or business logic. Agile PLM has an SSO mechanism that allows the PX to access the Agile server without the user having to re-authenticate. Agile passes encrypted SSO tokens that the PX then submits back to the Agile server. This token is a one-time token. Additionally, the token is secure as it is stored in the Agile Database and not accessible. This token is used to ensure that the SSO mechanism will be valid only once after the UI PX has been clicked by the user. Once the validation has been successful, the token will be removed from the secure place before providing the Agile server access to the PX. Finally, the token itself expires after a certain interval. The expiration time is configurable and Oracle recommends that customers keep this interval to a very small value to prevent misuse of this token.

Database-based Authentication

Customers can also use the Agile Database authentication, instead of the LDAP or SSO authentication mechanisms. For more information, see the "Account Policy" section in the *Agile PLM Administrator's Guide.*

Configuring and Using Access Control

Authorization primarily includes two processes:

1. Permitting only certain users to access, process, or alter data.

2. Applying varying limitations on user access or actions. The limitations placed on (or removed from) users can apply to objects, such as schemas, tables, or rows; or to resources, such as time (CPU, connect, or idle times).

Before creating a new Agile PLM user, make sure you answer the following questions:

□ What does this user need to be able to do in Agile PLM? What default roles are required for this user?

□ What should this user be prevented from doing in Agile PLM?

□ Will this user need to have separate Login and Approval passwords?

□ On which Agile PLM lists will the user's name appear?

□ Which Agile PLM searches should the user be able to use?

□ Is the user a Power User? A Power User can log in at any time and is not counted as a member of the concurrent user pool.

Do not assign too many users and designated escalation persons to user groups. Only assign users based on the requirements of each user group. Update user groups regularly.

For more information about access control using roles and privileges, see the *Agile PLM Administrator Guide.* Refer to the following relevant sections:

□ *Overview of Roles and Privileges in Agile PLM*

□ *Guidelines for Working with Roles*

□ *Securing and Maintaining Roles and Privilege Masks*

Configuring and Using Security Audit

Agile PLM allows you to audit your system by utilizing the User Monitor window, and through the data collected in an object's History Tab.

User Monitor

The User Monitor window lists the users that are presently logged in to the Agile PLM system. It displays the following information about each logged-in user.

Table column	Description
User Name	The first and last name of the logged in user.
User ID	The login username of the user.
Host	Indicates the user's host.
Login Time	The time the user logged in.

For more information, see the "User Monitor" section in the *Agile PLM Administrator Guide.*

History Tab

The History tab shows a summary of actions taken against an object. History is recorded for all objects in your Agile PLM system's database, and shows all actions by users and administrators. The History tab automatically populates.

The types of actions recorded for items are:

- Creation of the item

- Attachment actions: view, open, add, delete, get, check in, check out, cancel checkout, incorporate, unincorporate, and field modifications on the **Attachments** tab.

- Save As

- Send

- Print

- Modification of the subclass or any field of a released item

- Subscription modification and sharing

For more information see:

- *Getting Started with Agile PLM*

- *"History Tab" in the Agile Product Lifecycle Management Product Collaboration User Guide*

Log Files

An additional source of audit information is log files. You can enable logging controls in Agile or in the WebLogic Server so that you can get more security-related information.

For more information about enabling logging, refer to the section "Logging Configuration" in the *Agile PLM Administrator Guide.*

For more information about enabling logging scripts in WebLogic, see "Application Logging and WebLogic Logging Services" in the WebLogic Server documentation.

Security Considerations for Developers

This chapter includes the following:

This chapter discusses information useful to developers extending the application or producing applications using the product as a platform. Agile supports SDK if you prefer to use Java code for the extensions. Alternatively, Agile supports Web Services extensions so that you can use your preferred development language and platform.

Extensions Using Web Services

The Agile PLM application includes web services as an extensibility point. The out-of-box Agile PLM web services can be leveraged to provide customized clients or integration modules. Agile PLM web services authenticate using basic authentication.

For optimal security protection, Oracle recommends configuring the web services using SSL. For more information about how to configure SSL for web services, refer to Appendix B, SSL Configurations on page 23.

For more information about Agile web services, see *Agile Web Services User Guide*.

Extensions Using SDK

Oracle's Agile Software Development Kit (SDK) is a collection of Java application programming interfaces (APIs), sample applications, and documentation that enable you to build custom applications to access, or extend the functionalities of the Agile Application Server. Using the SDK, you can create programs that extend the functionality of the Agile PLM and can perform tasks against the PLM system.

For optimal security protection, use SSL. For more information about how to configure SSL for SDK, refer to Appendix B SSL Configurations on page 23.

For more information about SDK, in general, see the *SDK Developer Guide - Developing PLM Extensions*.

Chapter 7

Appendices

This chapter includes the following:

Secure Deployment Checklist

Follow the secure deployment checklist provided for the Oracle Database Server, as defined in the *Oracle Database Security Guide.* Similarly, follow guidelines for deploying your Oracle WebLogic Server, as defined in the Oracle WebLogic Server documentation.

The following security checklist includes guidelines that help secure your Agile PLM application:

1. Practice the principle of least privilege.

2. Enforce access controls effectively and authenticate clients stringently.

3. Restrict network access.

    1. Use a firewall.

    2. Never poke a hole through a firewall.

    3. Monitor who accesses your systems.

    4. Check network IP addresses.

    5. Encrypt network traffic.

    6. Harden the operating system.

4. Apply all security patches and workarounds.

5. Use strong passwords.

6. Deploy WebLogic Server using SSL.

7. Change the WebLogic administrator's username and password.

8. Set up a proxy server.

9. Contact the Oracle Security Support team if you come across any vulnerability in the Agile PLM application.

SSL Configurations

This Appendix includes the following:

You can set up SSL in your Agile PLM environment to work with the following:

- SDK

- Web Services

- WebLogic Server

- AutoVue Server

- Tomcat Server

Before you refer to the sections in this appendix to set up SSL for the specific environments, follow the steps described in

Basic SSL Configuration

Follow these basic steps for setting up SSL. The steps provide a general guideline on how to generate the SSL certificates before you can apply for the Agile PLM Application server and other Agile PLM components like SDK and WebService samples.

1. Generate and procure the security certificate for your setup as per your company's policies.

   Create a keystore file named like mykeystore.jks which could be located on any disk path and containing a private key and a self-signed public key.

   a. Go to the WLS domain directory and run setENV.cmd.

   b. Run the following keytool command under the path of the keystore that you want to generate.

   ```
   C:\CSR>keytool –genkey –keyalg RSA –alias mykey –keystore
   mykeystore.jks
   ```

2. Generate a Certificate Signing Request (CSR) and send it to Certifying Authority using the command: `C:\CSR>keytool –certreq –keystore mykeystore.jks`

The result should look similar to this example.

```
C:\CSR>keytool -certreq -keystore mykeystore.jks
Enter keystore password:
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIB1TCCAT4CAQAwgZQxCzAJBgNVBAYTAkNOMRAwDgYDVQQIEwdCZW1qaW5nMRAwDgYDVQQHEwdC
ZW1qaW5nMRswGQYDVQQKExJPcmFjbGUgQ29ycG9yYXRpb24xIjAgBgNVBAsTGUZPUiBURVNUUSU5H
IFBVU1BPU0UTIE9OTFkxIDAeBgNVBAMTF2J1ajMwMTI4My5jbi5vcmFjbGUuY29tMIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCm5PJ2emrfXE9U/tOjMFqYtXEFsLqZxiG0Kfe77SC1W5tyL3s8
7PgBMXY7snzlt1Bco+xOmujANLAC5uyNz8g/oFyNkhirrRjdqQnxp+9dyBMb2GI17UwRFpaP7E8k
fgpgFxvfdXt8WhosngOw9HLAceBH5vhs3EiruKfpMLN0nQIDAQABoAAwDQYJKoZIhvcNAQEFBQAD
gYEAJa2BGvyAK8PgbbFxEk8qKAverroC3/Nfwjhnn4arEEW2iHm4wuDIjBxINx8KJbHgGR74Pfy7
T6650N2D4U1TCXF99+19ODsZPOI5wAAJYui/EzTacs0ZF1P32gUGed+9HnKr/Yz4ws5/U1M/JdWQ
8dvPmJsp0L12MG/3ujxSBzo=
-----END NEW CERTIFICATE REQUEST-----

C:\CSR>_
```

3. The CA returns with the certificate reply, RootCA, and sometimes an intermediateCA certificate. Installing the newly issued certificate normally involves installing it along with its certificate trust chain, which basically means installing (or verifying prior installation of) the certificates of (a) VeriSign's public primary root CA (our trust anchor CA) and (b) of our issuing (intermediate) SSL CA before (c) your newly issued SSL certificate is installed.

In CA replying mails, we get the OraclePKI-SSLCA.zip file, newly issued SSL certificate

Assuming you get the certificates as a zip file, for example <Company>PKI-SSLCA.zip, it could contain the following set of certificates:

a. root CA certificate: VTN-PCA-3G3.pem (included in <Company>PKI-SSLCA.zip)

b. intermediate SSL CA certificate: <Company>_SSL_CA.pem (included in <Company>PKI-SSL CA.zip)

c. newly issued SSL certificate: at the bottom of the mail "Your Standard SSL Certificate Is Ready", copy"-----BEGIN CERTIFICATE----- … …-----END CERTIFICATE-----" into a text file, and save as a .pem file myCert.pem

4. Import the certificates into the keystore.

This can be done in two ways, either by importing the certificates in an order of RootCA, intermediateCA and then Certificate reply or by creating a certificate chain, clubbing them in order into a .pem file. In the following example, we create a certificate chain file CertChain.pem and import it into the identity keystore overriding the private key alias, which is mykey in this example.

It is important to remember that the certificate can only be installed on the server that already has its corresponding private key present in its keystore. Additionally, it must be the same private key that was created during the CSR generation process described above).

5. Open a text editor and paste the contents of each certificate. Paste the certificate C:myCert.pem, B:<Company>_SSL_CA.pem, A:VTN-PCA-3G3.pem from top to bottom and save as CertChain.pem.

```
-----BEGIN CERTIFICATE-----

C:<myCert.pem>

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

B:<<Company>_SSL_CA.pem>
```

```
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

A:<VTN-PCA-3G3.pem>

-----END CERTIFICATE-----
```

Our server should be presenting the whole chain of certificates to its clients during the SSL handshake. Certificates are typically presented in the leaf to root order (C-B-A).

Import the certificates chain to the keystore by running following command:

```
C:\CSR>keytool -import -file CertChain.pem -alias mykey -keystore
mykeystore.jks -storepass <password>
```

6. Create a trust keystore. Do this by importing your Root CA certificate into another keystore that constitutes the trust:

```
C:\CSR>keytool -import -file VTN-PCA-3G3.pem -alias rootca -keystore
trust.jks -storepass <password>
```

Configuring SSL on the WebLogic Server

Once you have completed the steps in Basic SSL Configuration on page 23, continue with the following procedures to configure SSL on the WebLogic Server that hosts the Agile PLM Application.

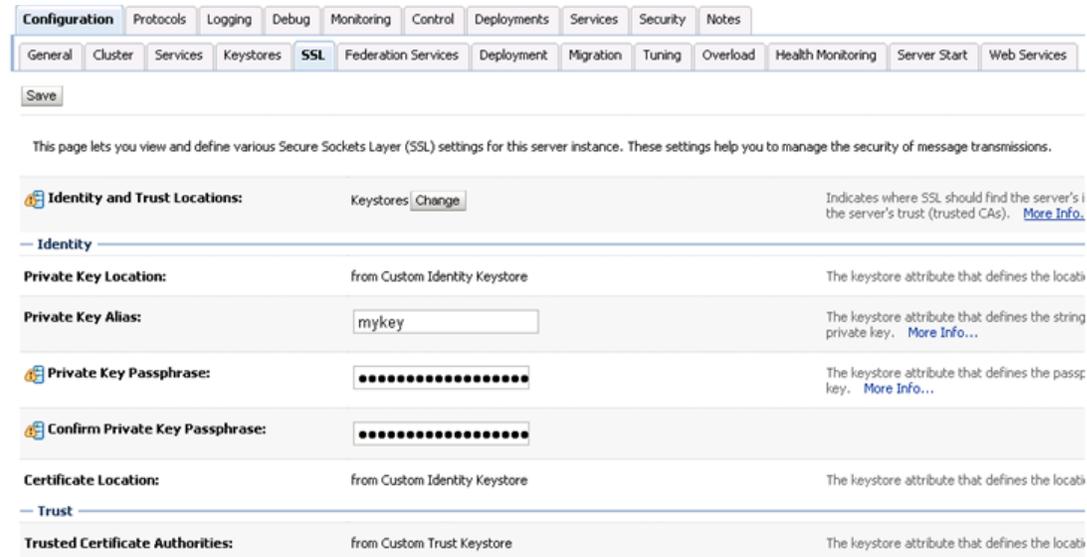### *Configuring the Keystore on the WebLogic Server*

1. Access http://<AgileApplicationServerName>:7001/console/login/LoginForm.jsp.

2. Log in to the Admin Console.

3. Select the server on which you want to configure the SSL certificate.

    Server –> Click on the Keystore tab. By default it points to the Demo Certificates.

4. From the dropdown list, select the "Custom Identity and Custom Trust" option. Enter the identity and trust keystore details.
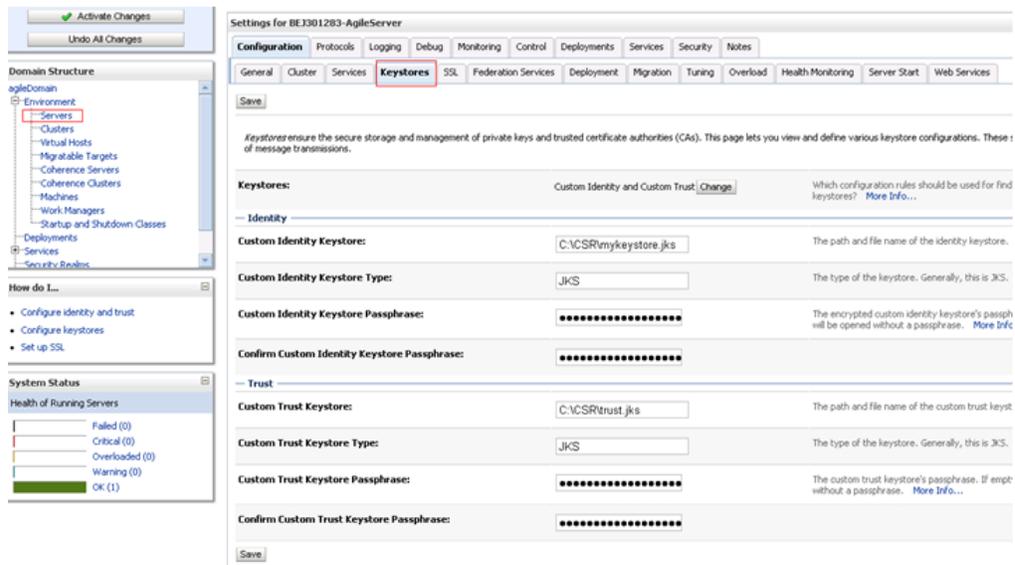
Configure the Identity of the Server

Click on the SSL tab and enter the alias of the private key, e.g. mykey, and the key passphrase "Agile123".

Enable SSL and Assign Port

Import client.p12 to IE

Agile PLM Application SSL Configuration

Modify the following configuration files:

1. **jndiurl.properties**

   E28772-01
   <AgileHomePath>\agileDomain\servers\<AgileServer>\tmp_WL_user\AgilePLM\<RandomName>\APP-INF\classes

   ```
   server1=t3s://<AgileApplicationServerName>:7002
   ```

2. **agile.properties**

   Path:
   ```
   <AgileHomePath>\agileDomain\servers\<AgileServer>\tmp\_WL_user\AgilePLM\<RandomName>\APP-INF\classes
   ```

   ```
   <AgileHomePath>\agileDomain\config
   ```

   ```
   ##### Common Web Security Settings    ###########
   ```

   ```
   # Specify whether to use the Secure flag to protect sensitive cookies
   ```
   ```
   WebSecurity.ForceSecureCookies = true
   ```

3. **ext.jnlp**

   Path:

```
<AgileHomePath>\agileDomain\servers\<AgileServer>\tmp\_WL_user\AgilePL
M\<RandomName>\JavaClient.war\wls
```

```
<jnlp spec="1.0+"
codebase="https://<AgileApplicationServerName>:7002/JavaClient">
```

4. **pcclient.jnlp**

   Path:
```
<AgileHomePath>\agileDomain\servers\<AgileServer>\tmp\_WL_user\AgilePL
M\<RandomName>\JavaClient.war\
```

```
<jnlp spec="1.0+"
codebase="https://<AgileApplicationServerName>:7002/JavaClient">
```

```
<argument>serverURL=t3s://<AgileApplicationServerName>:7002</argument>
```

Once you have completed modifying the configuration files, restart the application server to make the settings effective.

HTTPOnly and SecureFlag Flags in agile.properties

Whenever user-sensitive cookies are generated in Agile PLM, the HTTPOnly flag is also included in the Set-Cookie HTTP Response Header. This helps mitigate the risk of a client-side script accessing the protected cookie (if the browser supports it). You can change the value to 'false' in order to retain legacy behavior. From a secure system perspective, we recommend that customers keep this set to 'true'.

Additionally, Agile PLM does not mandate use of SSL, so setting the Secure flag will prevent non-SSL enabled customers from using Agile. The solution is to introduce a setting for secure mode and if enabled, then set the Secure Flag on all the sensitive cookies. This ensures that sensitive cookies are available in another application only via HTTPS. These cookies will not be available via HTTP, even if both the Agile PLM Application and the external application are deployed in the same domain. You can change the value to 'false' in order to retain legacy behavior. From a secure system perspective, we recommend that customers keep this set to 'true'.

Configuring SSL on the File Manager

Once you have completed the steps in Basic SSL Configuration on page 23, continue with the following procedures.

1. Add the following component to <AgileHomePath>\FileManager\conf\server.xml:

```
<Connector protocol="org.apache.coyote.http11.Http11Protocol"
    port="8443" maxThreads="200"
         scheme="https" secure="true" SSLEnabled="true"
      keystoreFile="<KeystorePath>" keystorePass="<KeystorePassword>"
      clientAuth="false" sslProtocol="TLS"/>
```

2. Configure the Java Client File Manager node as follows:

   Name: iFS

   API Name: IFS

   File Manager URL: https://<FileServerName>:8443/Filemgr/AttachmentServlet

Enabled: Yes

Primary file server: Yes

File Manager Internal Locator: https://<FileServerName>:8443/Filemgr/services/FileServer

Viewer Server URL: https://<FileServerName>:8443/Filemgr/VueServlet

Viewer Proxy URL: https://<FileServerName>:8443/Filemgr/VueLink

Viewer Content URL: https://<FileServerName>:8443/Filemgr/jVue

3. Configure `<AgileHomePath>\agileDomain\config\server.conf`

```
   app.server.url
=http://<AgileApplicationServerName>:7001/Agile/services/FSHelper

   file.server.url
=https://<FileServerName>:8443/Filemgr/services/FileServer

   dms.server.url =http://
<AgileApplicationServerName>:7001/Agile/services/DmsService
```

4. Restart file manager server.

5. Access https:// <FileServerName>:8443/Filemgr/Configuration to check the File Manager configuration.

Configuring SSL on the AutoVue 20.2

You can enable SSL communication between the AutoVue client and the VueServlet, as well as between the VueServlet and the AutoVue server. Detailed information on how to do the latter can be found in *Oracle AutoVue Client/Server Deployment Security Guide Release 20.2*.

Configuring SSL Between the AutoVue Client and the VueServlet

This section describes how to configure SSL between the AutoVue Client and the VueServlet. We do not provide steps for configuring SSL between the File Server and Auto Vue Server, because this communication takes place behind the firewall and SSL is not commonly used.

1. Connect to the application sever via HTTPS protocol in order get the application server's certificate, for example, https://<AgileApplicationServerName>.

2. Import the certificate into Internet Explorer.

3. Export the certificate from Internet Explorer as a base-64 encoded format and save the certificate onto the local disk, for example, C:\certs.cer.

4. Import the certificate into the AutoVue server's JRE using Java's keytool command:

```
<Java Install Directory>\bin>keytool -import -alias <servername> -file
c:\certs.cer -trustcacerts -v -keystore
C:\Oracle\AutoVue\jre\lib\security\cacerts
```

5. Restart the AutoVue server.

6. Configure the web page that embeds the AutoVue applet to point to the https:// URL for the VueServlet.

Configuring SSL for SDK

To configure SSL for SDK, do the following:

1. Get the certification key like mykeystore.jks, that is generated using the steps in Appendix B, and keep the mykeystore.jks file in a folder located on the machine where you want to run SDK, such as "C:\SDKSSL".

2. Follow these steps to run SDK sample code with an SSL environment:
    a. Download SDK sample files from OTN.
    b. Go to "..\SDK_AIS_Samples\sdk\samples\api\Login".
    c. Update URL, USERNAME and PASSWORD with SSL server information in Login.java.

        Set URL as https://hostname:port/Virtualpath
    d. Update run.bat
        -   Set JAVA_HOME & SDK_HOME
        -   Update Java Command: java -classpath .;c:\SSL\SDK\AgileAPI.jar -Djavax.net.ssl.trustStore=C:\SDKSSL\mykeystore.jks -Djavax.net.ssl.trustStorePassword=Agile123 Login
    e. Run run.bat.

Configuring SSL for Web Services

To configure SSL for Web Services, do the following:

1. Get the certification key, for example mykeystore.jks, that is generated using the steps in Appendix B, and keep the file in a folder located on the machine where you want to run Web Services, for example "C:\SSL".

2. Follow these steps to run Web Services sample code in an SSL environment:
    a. Download the Web Services sample files from OTN.
    b. Copy AgileAPI.jar file into "C:\SSL\WS\".
    c. Copy one of the sample Java files for creating an object, CreateObject.java in a location such as "C:\SSL\WS".
    d. Compile the file as follows:

        ```
        C:\SSL\WS>javac -classpath .;C:\SSL\WS\AgileAPI.jar
        CreateObject.java
        ```
    e. Run the sample as follows:

        ```
        C:\SSL\WS>java -classpath .;C:\SSL\WS\AgileAPI.jar -
        Djavax.net.ssl.trustStore=C:\SSL\mykeystore.jks -
        Djavax.net.ssl.trustStorePassword=Agile123 CreateObject
        ```

Note	These steps in this section do not make use of an Integrated Development Environment (IDE). Alternatively, you can also set up SSL for Web Services using an IDE.