Agile Product Lifecycle Management

Security Guide

Release 9.3.4

E52156-05

May 2018

Agile Product Lifecycle Management Security Guide, Release 9.3.4

E52156-05

Primary Author: Oracle Corporation

Contributing Author: Edlyn Sammanasu

Contributor:

Contents

5 Protecting Agile PLM Data

6 Configuring SSL

7 Enabling Security for Web Services

8 Disabling Security

A Secure Deployment Checklist

B Checklist for Configuring Web Services Security

C SSL Security Configurations for Developers

D WS Security Configurations for Developers

E SSL Protocol and Signature Algorithm Changes

Preface

Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain.

Audience

This document is intended for administrators and users of the Agile PLM products.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

Oracle's 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.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Document Scope

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:

- "Agile PLM Overview" on page 2-1 gives an overview of Agile PLM and its modules.
- "Overview of Security Fundamentals" on page 3-1 provides an overview of basic security principles which should be considered while setting up Agile PLM.
- "Performing a Secure Agile PLM Installation" on page 4-1 provides guidance on how to securely install the Oracle Database Server, Oracle Fusion Middleware Infrastructure, Agile PLM, and the Agile PLM Database.
- "Protecting Agile PLM Data" on page 5-1 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.
- "Configuring SSL" on page 6-1 and "Enabling Security for Web Services" on page 7-1 provide details on how to secure your environment SSL only or SSL and Web Services security.
- Appendix C, "SSL Security Configurations for Developers" and Appendix D, "WS Security Configurations for Developers" provide information needed for developers to extend the Agile PLM application or produce applications using Agile PLM as a platform.

2

Agile PLM Overview

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

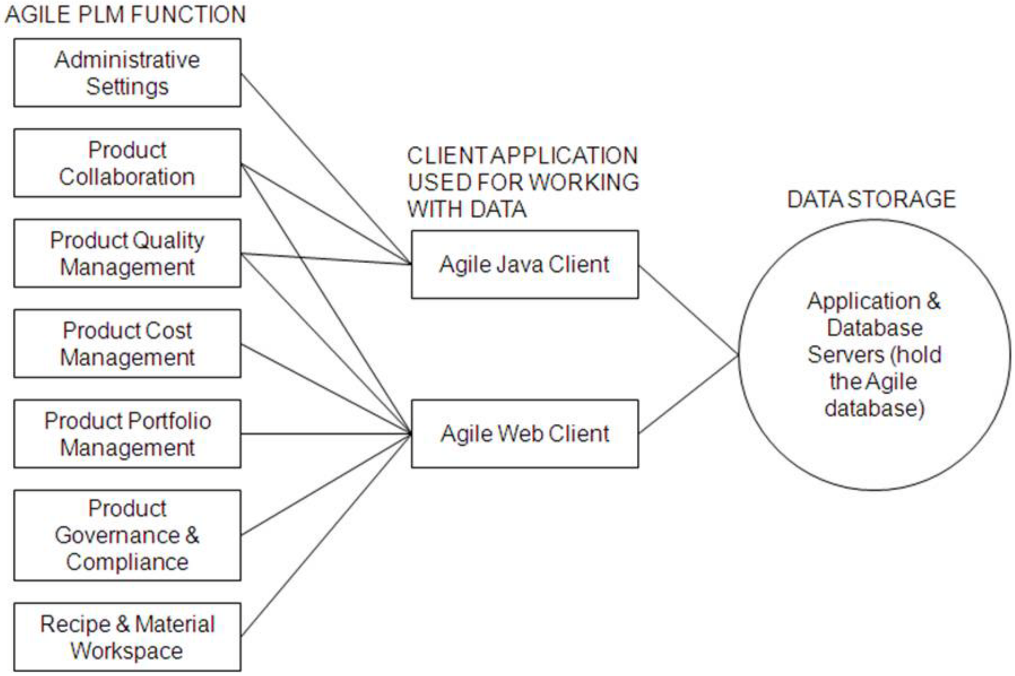
- **Agile Product Collaboration** (PC) - Management and collaboration of product record information throughout the product lifecycle, across internal organizations and the extended supply chain. Accessed through Web Client and Java Client.
- **Agile Product Governance & Compliance** (PG&C) - Management 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, and 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).

Figure 2–1 Relationships Between Components in an Agile Setup


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)

3
Overview of Security Fundamentals

This section describes some fundamental security principles and considerations.

Basic Security Considerations

The following general 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 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 can substitute for multiple, independent firewalls.

If you cannot use firewalls, then configure the TNS Listener Valid Node Checking feature. This feature restricts access based upon an IP address. Restricting database access by IP address, however, 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, and so on, especially early on in an organization's life cycle when people are few and work must 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.

4

Performing a Secure Agile PLM Installation

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. This chapter is a pre-requisite to the following chapters that describe security configuration procedures.

Understanding the Agile PLM Environment

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

- **Which resources must be protected?**
  - You must protect customer data, such as part numbers, file attachments, and so on
  - You must protect internal data, such as proprietary source code.
  - You must protect information in databases accessed by the Agile PLM server and the availability, performance, applications, and the integrity of the website.
  - You must protect system components from being disabled by external attacks or intentional system overloads.
- **Who are you protecting data from?**

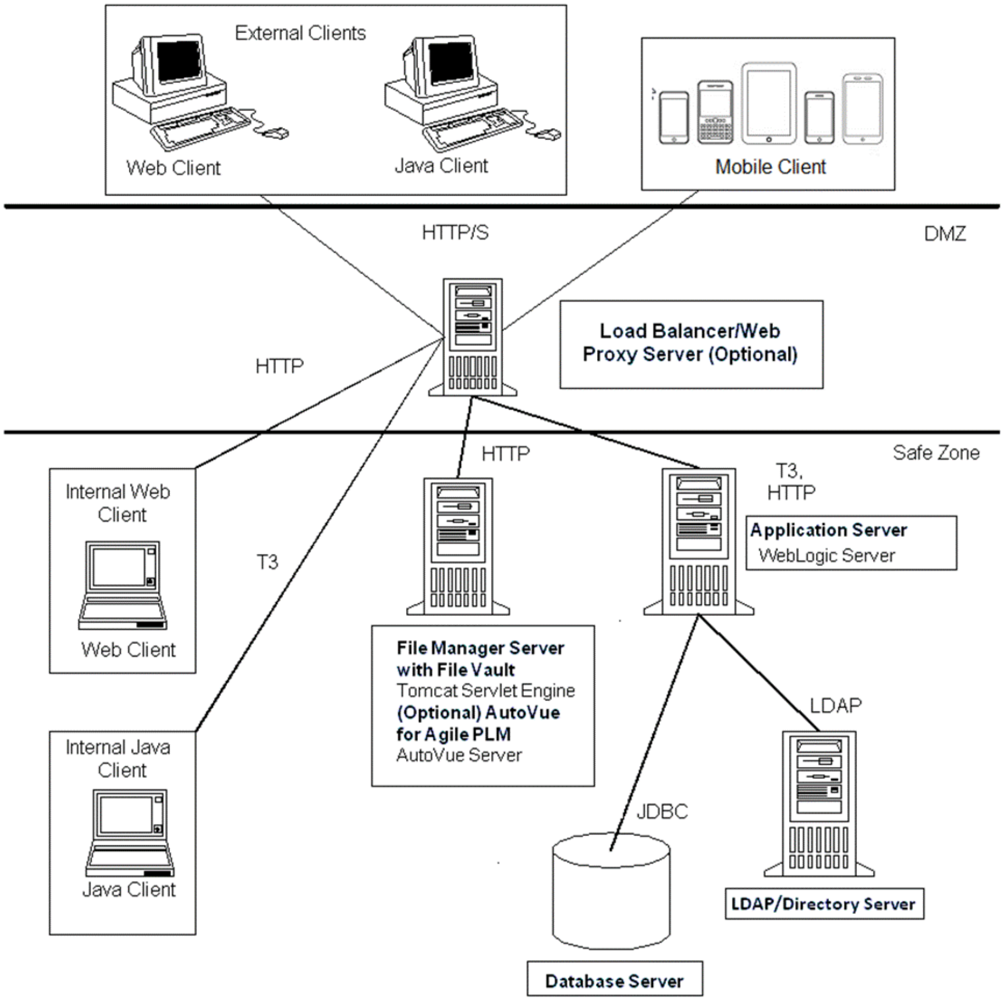
  For example, you must protect your subscribers' data from other subscribers, but someone in your organization might need to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, perhaps a system administrator can manage your system components 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 depicts the general topology that is recommended for a secure Agile PLM installation.

Figure 4–1 Recommended Deployment Topology



The components included in the topology diagram are defined below:

- **Agile PLM Clients** - Agile PLM includes three clients: a Web client, a Java client, and a Mobile client. The Web client is a thin HTML client that uses firewall-friendly protocols (HTTP/S). The Mobile client is a mobile application that also 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.

  Oracle recommends 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 figure, "Recommended Deployment Topology" on page 4-2 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.

  Oracle recommends communication using HTTP over SSL (HTTPS) for the most secure deployment. See Chapter 6, "Configuring SSL" for details on how to configure SSL for Agile PLM.

- **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, Oracle recommends 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. Oracle recommends 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 consists 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, and firewalls This document may be requested by Oracle Support if a network connectivity issue arises.

Determining Installation Flows Based on Security Needs

There are three installation flows that you can follow based on your security needs. They are as follows:

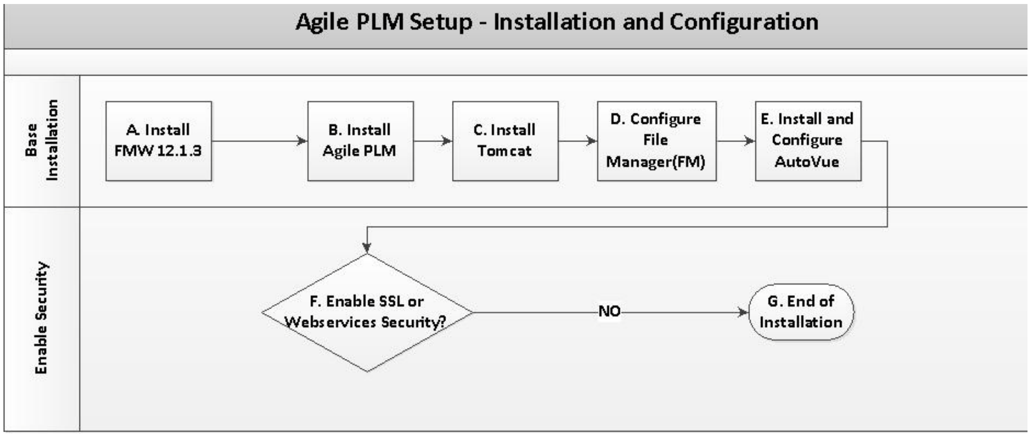
1. Agile installation without SSL or WSS enabled. See "Agile Setup Without SSL or WSS Enabled" on page 4-4
2. Agile installation with only SSL. See "Agile Setup With Only SSL Enabled" on page 4-4.

3. Agile installation with SSL and WSS enabled. See "Agile Setup With SSL and WSS Enabled" on page 4-5.

Agile Setup Without SSL or WSS Enabled

If you choose to not enable SSL or WSS, the basic process you need to follow is depicted in the following figure.

Figure 4–2 Agile PLM Setup Without SSL or WSS

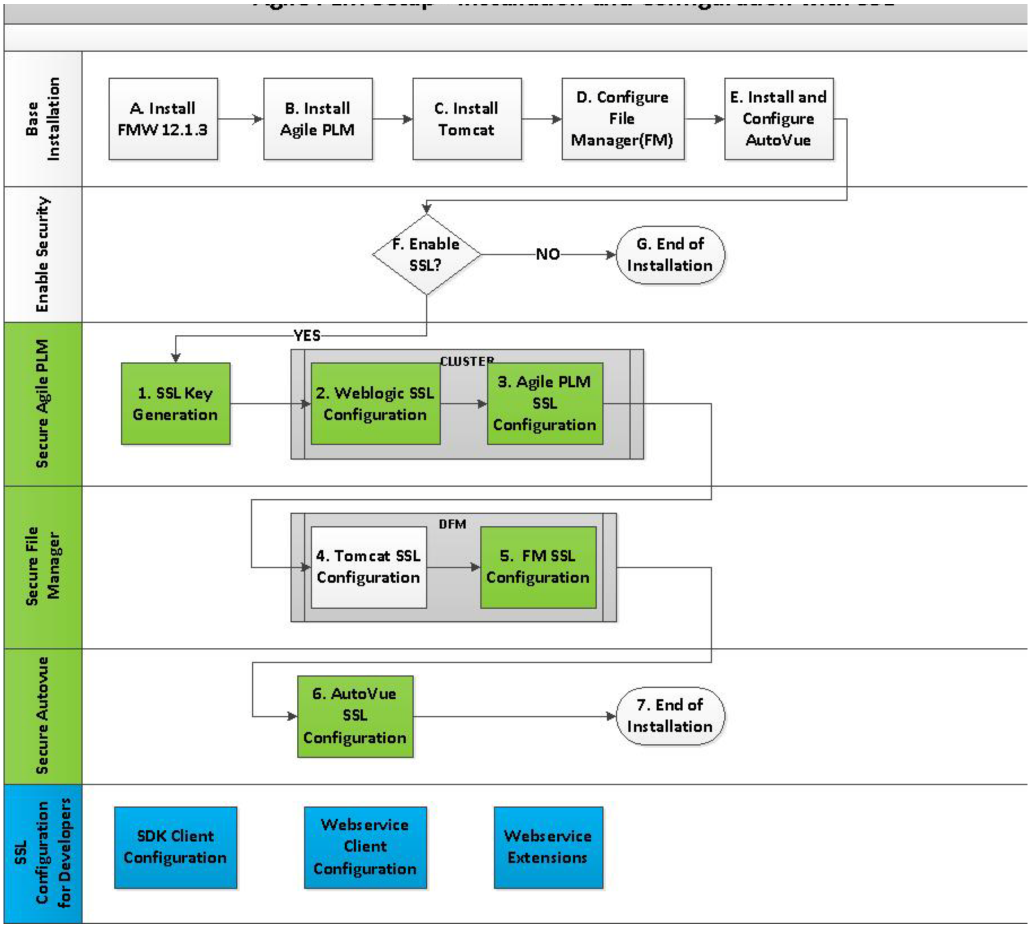

Agile PLM Setup - Installation and Configuration
Base Installation
Enable Security
A. Install FMW 12.1.3
B. Install Agile PLM
C. Install Tomcat
D. Configure File Manager(FM)
E. Install and Configure AutoVue
F. Enable SSL or Webservices Security?
NO
G. End of Installation

Refer to "Prerequisites Before Installing Agile PLM" on page 4-7 and "Installing Agile PLM" on page 4-7 for general guidance.

Agile Setup With Only SSL Enabled

If you choose to enable only SSL, the basic process you need to follow is depicted in the following figure.

Figure 4–3 Agile PLM Setup with Only SSL



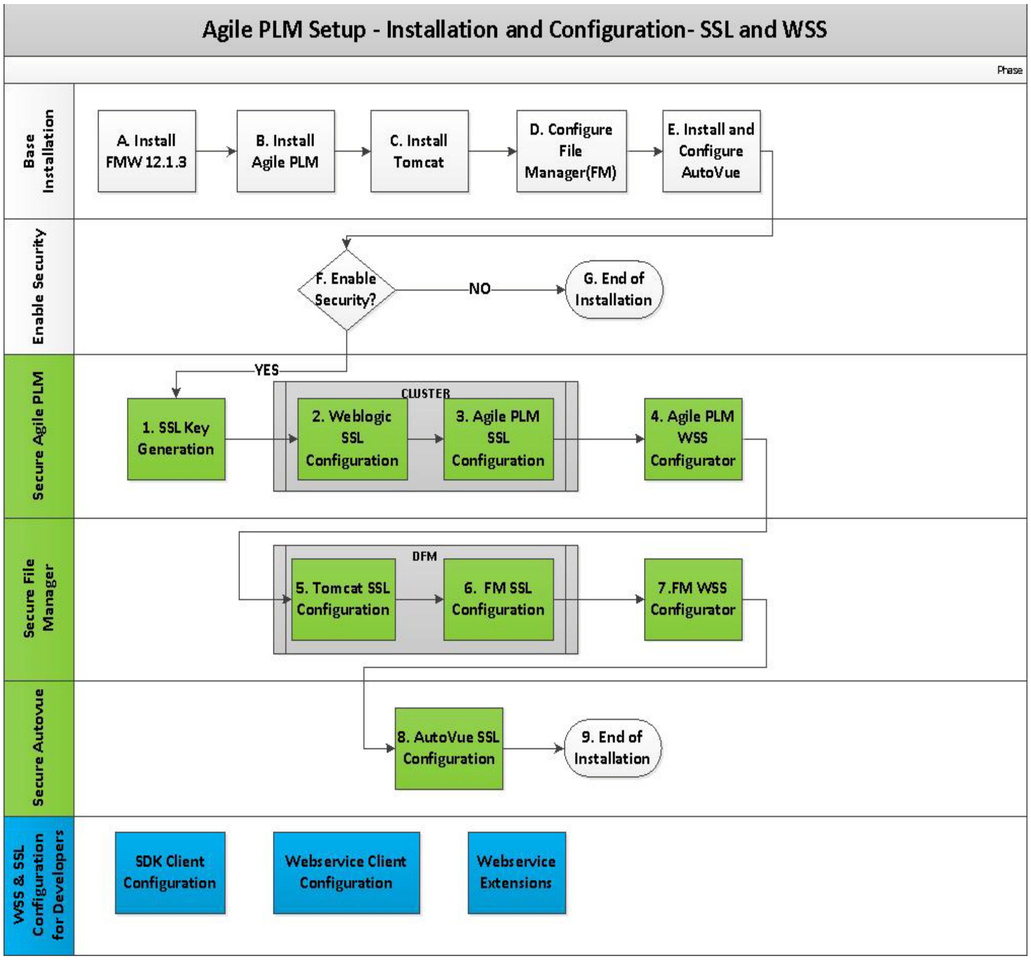
To set up only SSL, complete the general steps pictured above by referring to the sections mentioned below:

1. To complete steps A- E, refer to "Prerequisites Before Installing Agile PLM" on page 4-7 and "Installing Agile PLM" on page 4-7 for general guidance.
2. To complete steps 1-3, follow the relevant procedures in Chapter 6, "Configuring SSL", especially "Creating SSL Keystores and Truststore", "Configuring SSL on WebLogic Server" and "Configuring SSL in the Agile PLM Application".
3. To complete steps 4-5, follow the relevant procedures Chapter 6, "Configuring SSL", especially "Securing Agile PLM File Manager(s) Using SSL".
4. To complete step 6, follow the relevant procedures Chapter 6, "Configuring SSL", especially "Securing AutoVue 20.2.3 Using SSL".
5. To complete the WSS and SSL configurations needed by developers, refer to Appendix C, "SSL Security Configurations for Developers" and Appendix D, "WS Security Configurations for Developers".

Agile Setup With SSL and WSS Enabled

If you choose to enable both SSL and WSS, the basic process you need to follow is depicted in the following figure.

Figure 4–4 Agile PLM Setup with Both SSL and WSS


Agile PLM Setup - Installation and Configuration- SSL and WSS
Phase
Base Installation
A. Install FMW 12.1.3
B. Install Agile PLM
C. Install Tomcat
D. Configure File Manager(FM)
E. Install and Configure AutoVue
Enable Security
F. Enable Security?
NO
G. End of Installation
YES
Secure Agile PLM
1. SSL Key Generation
CLUSTER
2. Weblogic SSL Configuration
3. Agile PLM SSL Configuration
4. Agile PLM WSS Configurator
Secure File Manager
DFM
5. Tomcat SSL Configuration
6. FM SSL Configuration
7.FM WSS Configurator
Secure Autovue
8. AutoVue SSL Configuration
9. End of Installation
WSS & SSL Configuration for Developers
SDK Client Configuration
Webservice Client Configuration
Webservice Extensions

To set up SSL and WSS, complete the general steps pictured above by referring to the sections mentioned below:

1. To complete steps A- E, refer to "Prerequisites Before Installing Agile PLM" on page 4-7 and "Installing Agile PLM" on page 4-7 for general guidance.
2. To complete steps 1-3, follow the relevant procedures in Chapter 6, "Configuring SSL", especially "Creating SSL Keystores and Truststore", "Configuring SSL on WebLogic Server" and "Configuring SSL in the Agile PLM Application".
3. To complete step 4, follow the relevant procedures described in Chapter 7, "Enabling Security for Web Services", especially "Configuring WSS Policy for Agile PLM Web Services".
4. To complete steps 5-6, follow the relevant procedures Chapter 6, "Configuring SSL", especially "Securing Agile PLM File Manager(s) Using SSL".
5. To complete step 7, follow the relevant procedures described in Chapter 7, "Enabling Security for Web Services", especially "Configuring WSS Policy for File Manager Web Services".
6. To complete step 8, follow the relevant procedures Chapter 6, "Configuring SSL", especially "Securing AutoVue 20.2.3 Using SSL".

7. To complete the WSS and SSL configurations needed by developers, refer to Appendix C, "SSL Security Configurations for Developers" and Appendix D, "WS Security Configurations for Developers".

Prerequisites Before Installing Agile PLM

Before installing Agile PLM, you must install and configure Oracle Database Server and Oracle Fusion Middleware. 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 Fusion Middleware 12.1.3

For the latest information on how to install Oracle Fusion Middleware to setup WebLogic Server for Agile PLM, refer to the *Agile PLM Application Installation Guide*.

Installing Agile PLM

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

For the latest information on installing the File Manager, refer to the *Agile PLM Application Installation Guide.*

Optional Component Configuration

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

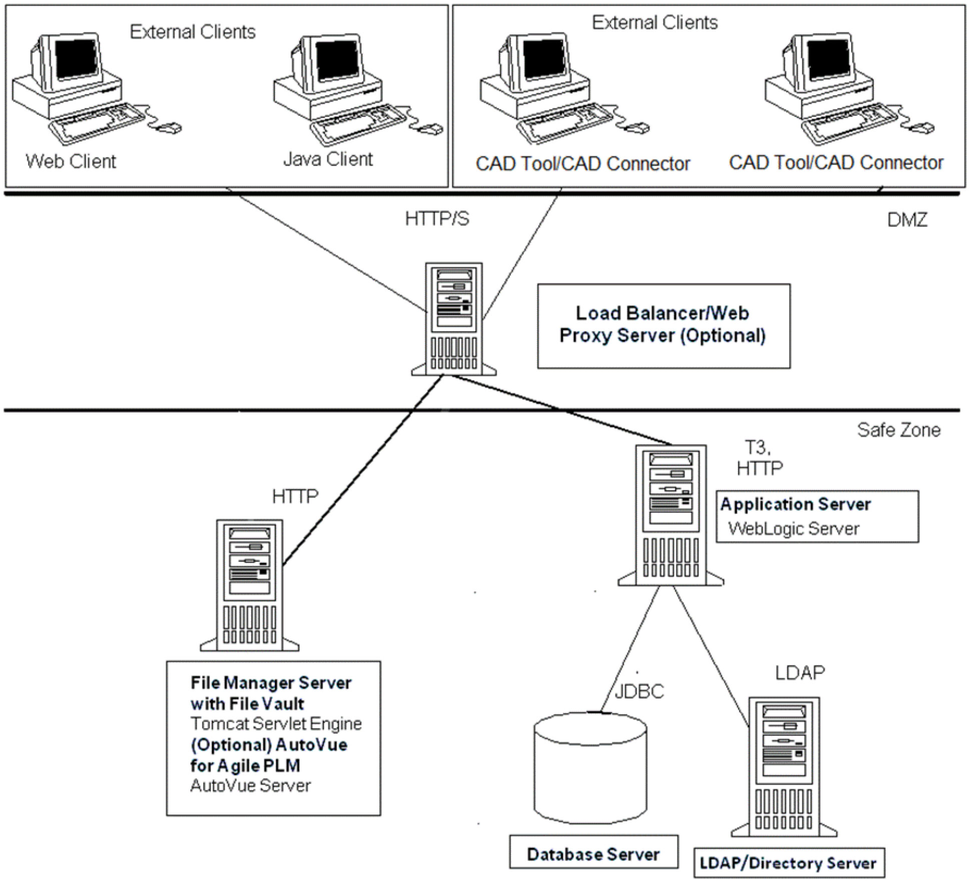
Configuring AutoVue (Optional)

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

Configuring MCAD Connectors (Optional)

The following diagram depicts how Oracle recommends that every CAD Tool/Connector be set up for optimal security.

Figure 4–5 Recommended MCAD Configuration



Oracle recommends that you configure the Engineering Collaboration Clients with HTTP(S). Refer to the "Configuring Engineering Collaboration Clients for HTTPS" section in the *MCAD Connectors for Agile Engineering Collaboration Administration Guide* for information about configuring MCAD Connectors securely.

5

Protecting Agile PLM Data

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 entered password 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

- Oracle Virtual Directory

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) - The secure configuration practices to configure Agile PLM with OAM Server can be found here: `http://fusionsecurity.blogspot.co.uk/2010/04/security-clarification-oam-identity.html.`
- 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 is valid only once after the UI PX has been clicked by the user. Once the validation has been successful, the token is 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 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 enables 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 gets automatically populated.

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 the set of 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.

6 Configuring SSL

This chapter describes how to configure SSL in Agile PLM, in Agile PLM File Manager(s), and AutoVue.

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

- SDK
- Web Services
- WebLogic Server
- Tomcat Server
- AutoVue Server

WARNING: Once you enable SSL for one of the components listed in the previous step, you must enable SSL for all components listed.

Before you refer to the sections in this appendix to set up SSL for any specific environment, follow the steps described in the following section, *Basic SSL Configuration*.

Note: For instructions on how to mitigate vulnerabilities related to SSL 3.0, see Appendix E, "SSL Protocol and Signature Algorithm Changes".

Tip: If you are planning on configuring SSL and Web Services Security, use the checklist in Appendix B, "Checklist for Configuring Web Services Security" to help keep track of your progress.

Creating SSL Keystores and Truststore

In the following sections, to configure SSL in your Agile PLM environment, you will need to refer to this section to generate a separate SSL key for each of the following components:

- WebLogic Server
- Tomcat Server

To generate the SSL keystore for File Manager, do the following:

1. Create a directory, C:\fm-keystore-934, then open the Command Prompt and navigate to this folder.

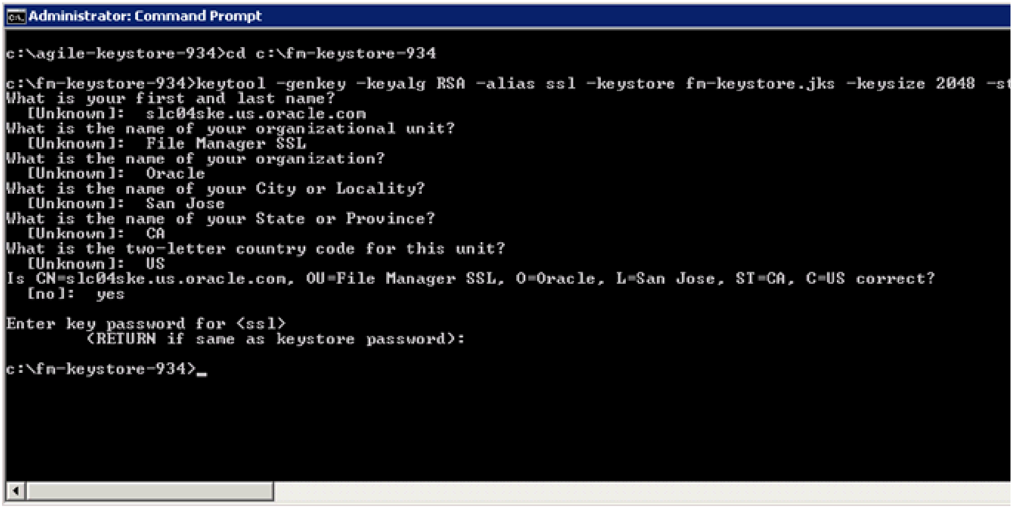
2. Execute the command:

   keytool -genkey -keyalg RSA -alias ssl -keystore fm-keystore.jks -keysize 2048 -storepass agile123

3. Enter the following information for the key generation command: CN=<File Manager hostname>, OU=Agile PLM, O=Oracle, L=San Jose, ST=CA, C=US. For example, CN=slc04ske.us.oracle.com, OU=Agile PLM, O=Oracle, L=San Jose, S=CA, C=US

4. Press Enter. This creates the fm-keystore.jks in c:\fm-keystore-934 directory.

Figure 6–1



```
c:\agile-keystore-934>cd c:\fm-keystore-934

c:\fm-keystore-934>keytool -genkey -keyalg RSA -alias ssl -keystore fm-keystore.jks -keysize 2048 -st
What is your first and last name?
  [Unknown]:  slc04ske.us.oracle.com
What is the name of your organizational unit?
  [Unknown]:  File Manager SSL
What is the name of your organization?
  [Unknown]:  Oracle
What is the name of your City or Locality?
  [Unknown]:  San Jose
What is the name of your State or Province?
  [Unknown]:  CA
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=slc04ske.us.oracle.com, OU=File Manager SSL, O=Oracle, L=San Jose, ST=CA, C=US correct?
  [no]:  yes

Enter key password for <ssl>
        (RETURN if same as keystore password):

c:\fm-keystore-934>_
```

To generate the SSL keystore and truststore for WebLogic, do the following:

1. Create a directory, C:\agile-keystore-934, then open the Command Prompt and navigate to this folder.

2. To generate a key in file agile-keystore.jks, execute the following command:

   ```
    keytool -genkey -keyalg RSA -alias ssl -keystore agile-keystore.jks
   -keysize 2048 -storepass agile123
   ```

3. Enter the following information for the key generation command:

   CN=<Application Server hostname>, OU=Agile PLM, O=Oracle, L=San Jose, ST=CA, C=US.

   For example, CN=myapp.mydomain.com, OU=Agile PLM, O=Oracle, L=San Jose, S=CA, C=US

4. Press Enter. The file agile-keystore.jks should now exist in directory C:\agile-keystore-934.

5. To export the SSL certificate from agile-keystore.jks, execute the following command:

   ```
    keytool -export -file agile-ssl-cert.cer -keystore agile-keystore.jks
   -storepass agile123 -alias ssl
   ```

6. To import the SSL certificate into file agile-trust-store.jks, execute the following command:

   ```
   keytool -import -file agile-ssl-cert.cer -keystore
   agile-trust-store.jks -storepass agile123
   ```

7. Press Enter. The file agile-trust-store.jks should now exist in directory C:\agile-keystore-934.

You need two JKS keystores for the WebLogic SSL configuration. One is the identity keystore, for example, agile-keystore.jks, and the other is the trust keystore, for example, agile-trust-store.jks. Leverage the Java Keytool to export the public key from the identity keystore into the trust keystore. For FM/DFM SSL configuration, you only need a single keystore, for example,.fm-keystore.jks.

> **Note:** If you use a CA certificate, you need to export the CSR and send it to CA, then you will receive the (A) root CA certificate, (B) sometimes intermediate certificates, and (C) CA signed certificate. You then need to import the signed certificates in the order of A-B-C back to the keystore that contains your private key. For WebLogic SSL configuration, you also need to import these certificates into the trust keystore, for example, agile-trust-store.jks.

Securing Agile PLM Application Using SSL

The following sections describe how to enable SSL for security in Agile PLM.

> **Note:** When SSL is enabled, you must ensure that the agileDomain\bin\startAgile.cmd | .sh WebLogic server file is protected using File Access Permissions. Visibility/accessibility should be limited to only users with root or elevated privileges. This file contains sensitive password data.

Configuring SSL on WebLogic Server

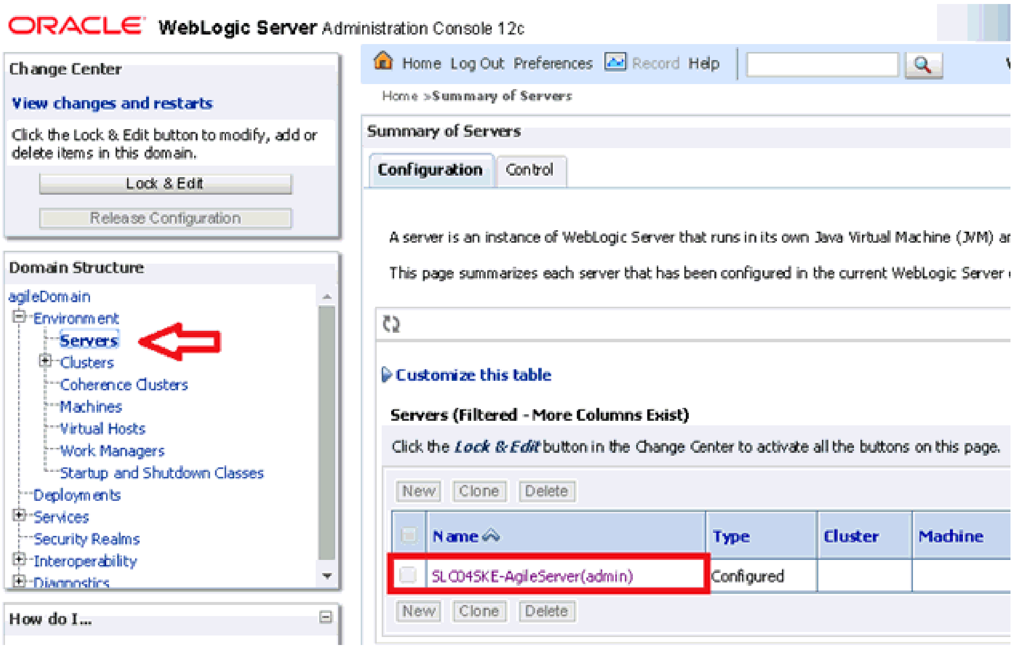
Complete the steps in "Creating SSL Keystores and Truststore" on page 6-1 and then continue with the following procedures to configure SSL on the WebLogic Server that hosts the Agile PLM Application.

Configuring the Keystore on the Weblogic Server

To configure the Keystore:

1. In a browser, launch http://<AgileApplicationServerName>:7001/console/login/LoginForm.jsp.
2. Log in to the Admin Console.
3. Expand Environment, click on Servers, and click on the server name on the right panel.

Figure 6–2



Click the Lock & Edit button to modify, add or delete items in this domain.

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) ar

This page summarizes each server that has been configured in the current WebLogic Server

Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

	Name	Type	Cluster	Machine
□	SLC04SKE-AgileServer(admin)	Configured		

4. In AgileServer > Configuration > Keystores, use Custom Idenitity and Custom Trust for keystores.

   - In the Identity Section provide the following:

     – Location in Custom Identity Keystore field: C:\agile-keystore-934\agile-keystore.jks

     – Enter "JKS" as the Custom Identity Keystore Type.

     – Enter "agile123" as the Custom Identity Keystore Passphrase.

   - In the Trust Section provide the following:

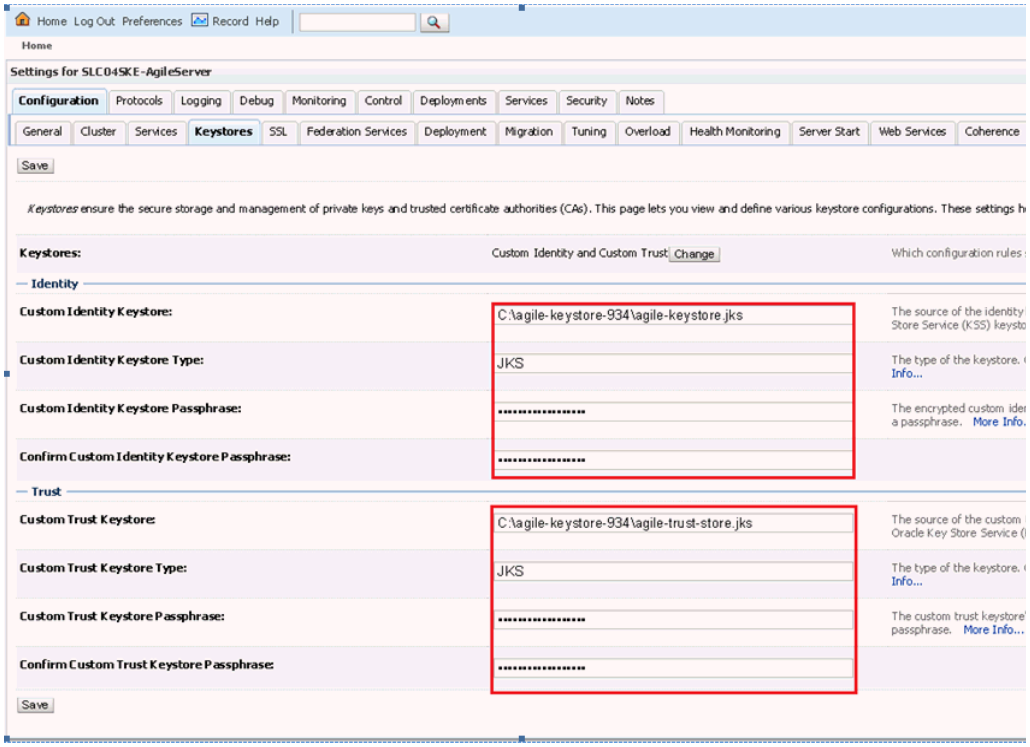
     – Location in Custom Trust Keystore field: C:\agile-keystore-934\agile-trust-store.jks

     – Enter "JKS" as the Custom Keystore Type.

     – Enter "agile123" as the Custom Trust Keystore Passphrase

   - Click Save.

Figure 6–3



Home Log Out Preferences Record Help
Home

Settings for SLC04SKE-AgileServer

Configuration | Protocols | Logging | Debug | Monitoring | Control | Deployments | Services | Security | Notes
General | Cluster | Services | Keystores | SSL | Federation Services | Deployment | Migration | Tuning | Overload | Health Monitoring | Server Start | Web Services | Coherence
Save

Keystores ensure the secure storage and management of private keys and trusted certificate authorities (CAs). This page lets you view and define various keystore configurations. These settings h

Keystores: Custom Identity and Custom Trust Change — Which configuration rules

Identity

Custom Identity Keystore:	C:\agile-keystore-934\agile-keystore.jks	The source of the identity Store Service (KSS) keysto
Custom Identity Keystore Type:	JKS	The type of the keystore. Info...
Custom Identity Keystore Passphrase:	••••••••••••••••••	The encrypted custom ider a passphrase. More Info.
Confirm Custom Identity Keystore Passphrase:	••••••••••••••••••	

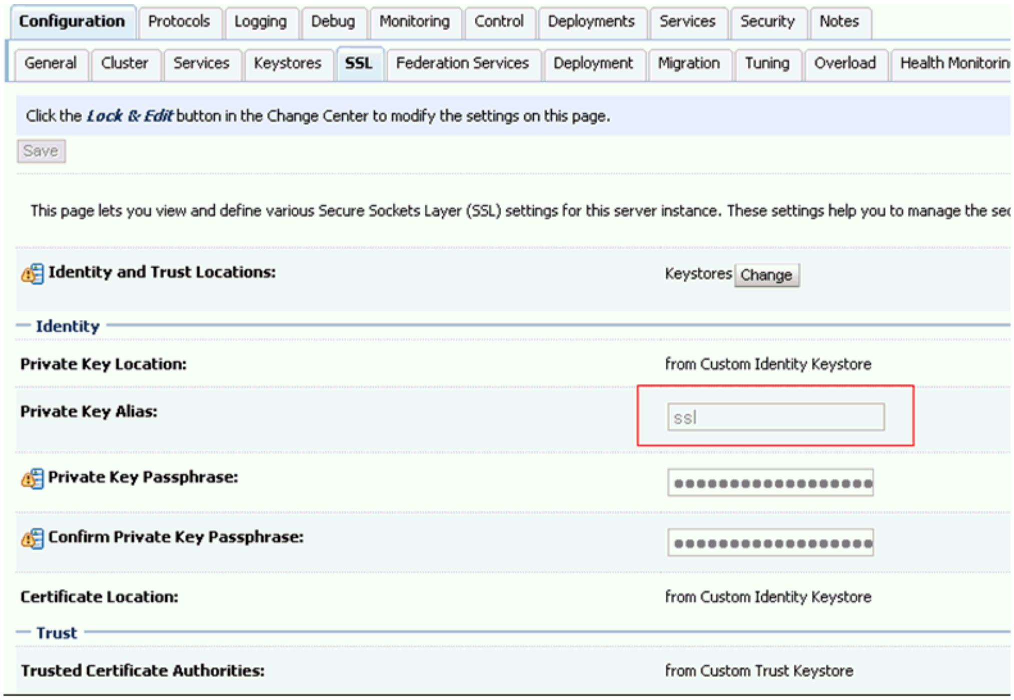
Trust

Custom Trust Keystore:	C:\agile-keystore-934\agile-trust-store.jks	The source of the custom Oracle Key Store Service (
Custom Trust Keystore Type:	JKS	The type of the keystore. Info...
Custom Trust Keystore Passphrase:	••••••••••••••••••	The custom trust keystore' passphrase. More Info...
Confirm Custom Trust Keystore Passphrase:	••••••••••••••••••	

Configuring the Identity of the WebLogic Server

1. Go to AgileServer > Configuration > SSL as shown in the following figure. In this example, "ssl" is the key, and the password is "agile123".

Figure 6–4



Click the ***Lock & Edit*** button in the Change Center to modify the settings on this page.

This page lets you view and define various Secure Sockets Layer (SSL) settings for this server instance. These settings help you to manage the sec

Identity and Trust Locations: Keystores

Identity

Private Key Location: from Custom Identity Keystore

Private Key Alias: ssl

Private Key Passphrase: ●●●●●●●●●●●●●●●●●●●

Confirm Private Key Passphrase: ●●●●●●●●●●●●●●●●●●●

Certificate Location: from Custom Identity Keystore

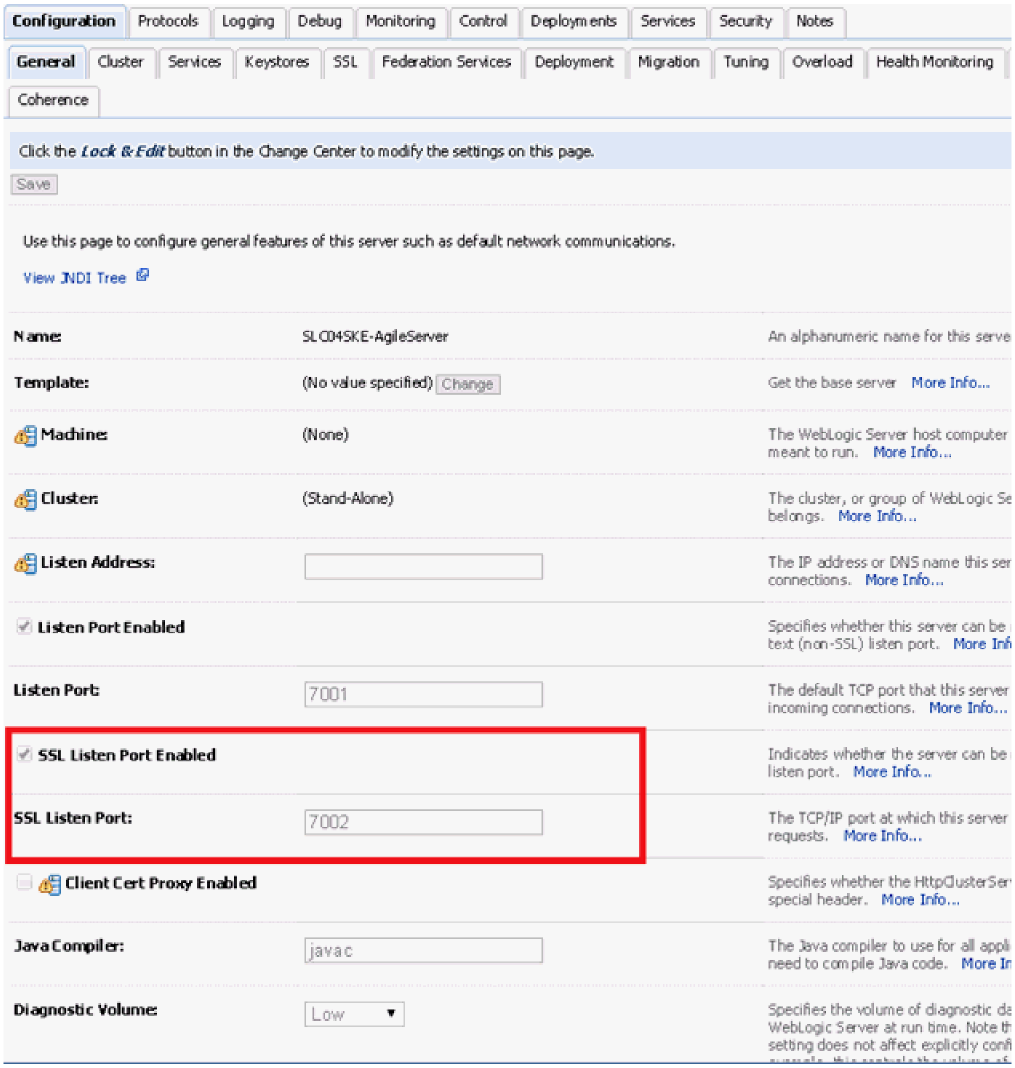
Trust

Trusted Certificate Authorities: from Custom Trust Keystore

Configuring SSL Listen Port for WebLogic Server

1. Navigate to AgileServer > Configuration > General and select the SSL Listen Port Enabled checkbox. The default SSL port is 7002.

Figure 6–5



Click the *Lock & Edit* button in the Change Center to modify the settings on this page.

Use this page to configure general features of this server such as default network communications.

View JNDI Tree

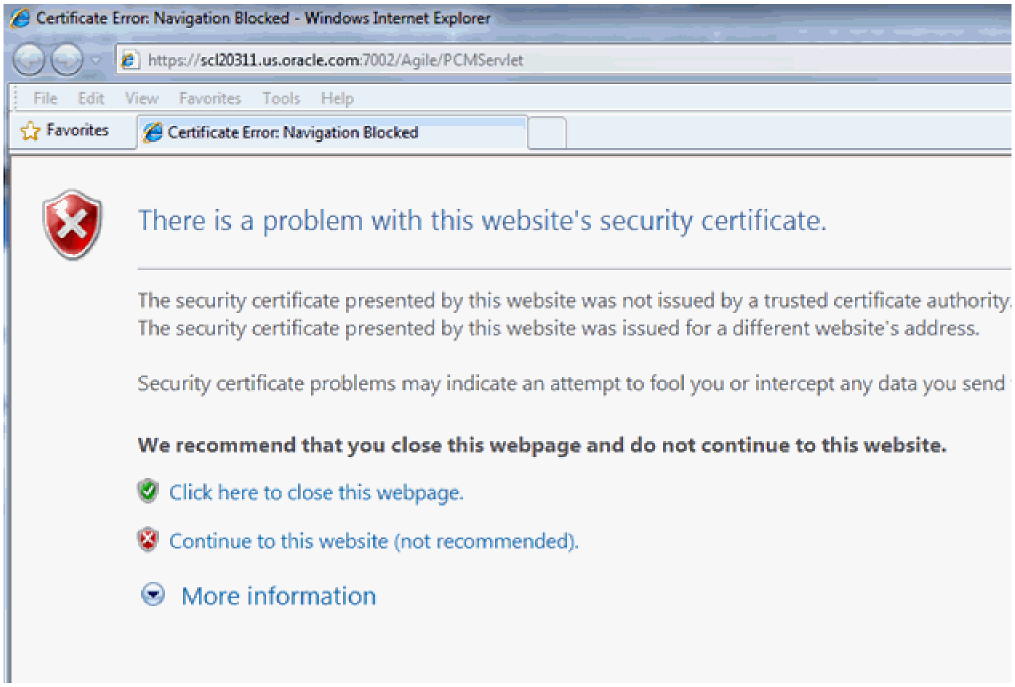
Name:	SLC04SKE-AgileServer	An alphanumeric name for this serve
Template:	(No value specified)	Get the base server More Info...
Machine:	(None)	The WebLogic Server host computer meant to run. More Info...
Cluster:	(Stand-Alone)	The cluster, or group of WebLogic Se belongs. More Info...
Listen Address:		The IP address or DNS name this ser connections. More Info...
☑ Listen Port Enabled		Specifies whether this server can be text (non-SSL) listen port. More Inf
Listen Port:	7001	The default TCP port that this server incoming connections. More Info...
☑ SSL Listen Port Enabled		Indicates whether the server can be listen port. More Info...
SSL Listen Port:	7002	The TCP/IP port at which this server requests. More Info...
□ Client Cert Proxy Enabled		Specifies whether the HttpClusterSer special header. More Info...
Java Compiler:	javac	The Java compiler to use for all appli need to compile Java code. More In
Diagnostic Volume:	Low	Specifies the volume of diagnostic da WebLogic Server at run time. Note th setting does not affect explicitly confi

2. Click Save to activate the changes in WebLogic Console.

Verify SSL Configuration on WebLogic Server

1. Connect to https://<hostname>:7002/Agile/PCMServlet and click Continue

Figure 6–6



Certificate Error: Navigation Blocked - Windows Internet Explorer

https://scl20311.us.oracle.com:7002/Agile/PCMServlet

There is a problem with this website's security certificate.

The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

2. Click the Certificate Errors button in URL bar. (The button is only available, if you used a self-signed certificate.)
3. View Certificates.
4. Log in to Agile.

The SSL setup is now complete and running on your WebLogic server.

Cluster Environment: Additional Configurations

You need to configure SSL for each WLS server in the cluster. You also need to configure SSL on Load Balancer (LB), and update the LB URI into Agile PLM Application SSL Configurations. Meanwhile, you have to import the LB SSL certificate into the trust keystore for every WLS server, and import all the WLS server's SSL certificates into LB trust keystore.

Configuring SSL in the Agile PLM Application

Modify the following configuration files for the SSL environment:

1. **jndiurl.properties**

   Path: <Agile_Home>\agileDomain\application\application.ear\APP-INF\classes

   server1=t3s://<app_server_alias>:7002

2. **agile.properties**

   Path: <Agile_Home>\agileDomain\config

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

   ```
   # Specify whether to use the Secure flag to protect sensitive cookies
   ```

```
WebSecurity.ForceSecureCookies = true
```

3. **ext.jnlp**

   Path:
   <AgileHome>\agileDomain\application\application.ear\JavaClient.war\wls

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

4. **pcclient.jnlp**

   Path: Agile_Home>\agileDomain\application\application.ear\JavaClient.war

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

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

   ```
   <argument>jvuecodebase=https://<fm_server_
   alias>:8443/Filemgr/jVue</argument>
   ```

   ```
   <argument>jvueserver=https://<app_server_
   alias>:7002/Agile/VueServlet</argument>
   ```

5. **custom.jnlp**

   Path: <Agile_Home>\agileDomain\application\application.ear\JavaClient.war

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

6. **startAgile.cmd |.sh**

   Path: <Agile_Home>\agileDomain\bin\startAgile.cmd |.sh

   Append the following two parameters to JAVA_OPTIONS:

   ```
   -Djavax.net.ssl.trustStore=C:\agile-keystore-934\agile-trust-store.jks
   -Djavax.net.ssl.trustStorePassword=Agile123
   ```

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 flag's value to false to retain legacy behavior. From a secure system perspective, however, Oracle recommends that customers keep the HTTPOnly flag set to true.

Additionally, Agile PLM does not mandate use of SSL, so setting the Secure flag prevents 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 through HTTPS. These cookies are not available through 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 to retain legacy behavior. From a secure system perspective, however, Oracle recommends that customers keep this flag set to true.

Securing Agile PLM File Manager(s) Using SSL

The following section describes how to configure SSL on a File Manager.

> **Note:** When SSL is enabled, you must ensure that the FileManager\conf\server.xml Tomcat server file is protected using File Access Permissions. Visibility/accessibility should be limited to only users with root or elevated privileges. This file contains sensitive password data.

Configuring SSL on the Tomcat Server

Complete the steps in "Creating SSL Keystores and Truststore" on page 6-1 and then continue with the following procedures.

1. Export the File Manager SSL certificate by executing following command:

   ```
   keytool -export -file fm-ssl-cert.cer -keystore fm-keystore.jks
   -storepass agile123 -alias ssl
   ```

   The fm-keystore-934 folder now contains the fm-ssl-cert.cer file and the fm-keystore.jks file.

2. Copy the C:\agile-keystore-934\agile-ssl-cert.cer to C:\fm-keystore-934 folder.

3. Import agile-ssl-cert.cer into File Manager key store.

   keytool -import -file agile-ssl-cert.cer -alias a9-ssl-cert -keystore fm-keystore.jks -storepass agile123

4. Copy C:\fm-keystore-934\fm-ssl-cert.cer to C:\agile-keystore-934 folder.

5. In the command prompt, change directory to C:\agile-keystore-934.

6. Import FM SSL certificate into WLS SSL Trust Keystore using following command:

   ```
   keytool -import -file fm-ssl-cert.cer -alias fm-ssl-cert -keystore
   agile-trust-store.jks -storepass agile123
   ```

7. Open <AgileHomePath>\FileManager\conf\server.xml and add a new connector. The file manager SSL port is 8443. Place the connector code after the code for the connector of port 8080, as shown in the following example:

   ```
   <Connector port="8080" protocol="HTTP/1.1"
   connectionTimeout="20000"redirectPort="8443" />

   <Connector protocol="org.apache.coyote.http11.Http11Protocol"
   port="8443" maxThreads="200"

   scheme="https" secure="true" SSLEnabled="true"
   keystoreFile="C:\fm-keystore-934\fm-keystore.jks"
   keystorePass="agile123" keyAlias="ssl"clientAuth="false"
   sslProtocol="TLS"/>
   ```

8. To configure SSL on the File Manager application, change <AgileHomePath>\agileDomain\config\server.conf as following:

   ```
   app.server.url=https://<app_server_
   alias>:7002/Agile/FSHelper/FSHelperWSService

   file.server.url=https://<fm_server_
   alias>:8443/Filemgr/services/FileServer
   ```

```
dms.server.url=https://<app_server_
alias>:7002/Agile/DmsService/DmsViewerAPIService
```

9. To configure the Java Client File Manager node, log in to Java Client, navigate to Admin >Server Settings > Locations, and do the following:

   - Change General Information > Web Server URL to https://<app_server_alias>:7002/Agile/PLMServlet
   - Change Java Client URL to https://<app_server_alias>:7002/JavaClient/start.jsp
   - Change File Manager > iFS to https://<fm_server_alias>:8443/Filemgr/AttachmentServlet

10. To configure SSL on DFM Tomcat, do the following:

    - Set up DFM.
    - Copy fm-keystore-934 folder from FileManager and place it on C:\ drive of DFM.
    - Change server.xml of Tomcat to enable SSL.
    - Remove SSL key from fm-keystore.jks of DFM as its for File Manager.
    - Generate New SSL key for DFM.
    - Export DFM SSL cert.
    - Import DFM SSL cert into A9 trust store (agile-trust-store.jks) and File Manager keystore (fm-keystore.jks).

11. Restart the file manager server and access https:// <fm_server_alias>:8443/Filemgr/Configuration to check the File Manager configuration.

SSL is now configured on File Manager. Restart the File Manager and it should work as expected.

Securing AutoVue 20.2.3 Using SSL

Note: You may restrict IP access between the VueLink/VueServlet and the AutoVue Server. For details, refer to the Security Features section "Configuring and Using Restrict IP Access" in the *Oracle AutoVue Integration Software Development Toolkit Security Guide*.

Configuring SSL Between the AutoVue Server and the VueServlet

Note: The VueServlet is hosted by File Manager, therefore, configuring File Manager for SSL also configures the VueServlet URL for SSL.

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

1. Connect to the File Manager Server through HTTPS protocol, https://<fm_server_alias>:8443/Filemgr/Configuration.

2. Import the certificate into Internet Explorer.
3. Export the certificate from IE as base-64 encoded format and save the certificate onto the local disk, for example, C:\fmssl.cert
4. Import both the WebLogic server and Tomcat server certificates into the AutoVue Server's JRE using Java's keytool command:

   ```
   <Java Install Directory>\bin>keytool -import -alias <appservername>
   -file C:\cert.cer -trustcacerts -v keystore
   <AutoVueHome>\jre\lib\security\cacerts
   ```

   ```
   <Java Install Directory>\bin>keytool -import -alias <fmservername>
   -file C:\fmssl.cer -trustcacerts -v keystore
   <AutoVueHome>\jre\lib\security\cacerts
   ```

5. Restart the AutoVue server.

7

Enabling Security for Web Services

Agile PLM leverages Oracle Web Service Management (OWSM), which provides a policy framework to manage and secure A9 web services on WebLogic server, and the Metro Security framework to secure A9 web services on Tomcat server. Agile PLM provides a convenient configuration tool, WSSConfigurator, for you to configure WSS policies for A9 web services deployed on both WebLogic and Tomcat servers.

Note: FMW Patch 20020473 is required as a prerequisite.

To set up Web Services Security, you have to do the following:

- Ensure that OWSM is installed in the Agile domain.
- Configure WSS Policy for Agile PLM Web Services.
- Configure WSS Policy for File Manager Web Services.

Follow the procedures in the this chapter to set up Web Services Security in your environment.

Installing OWSM on the Agile Domain

By default, the OWSM component is not installed in Agile Domain. The WSSConfigurator tool reminds users to extend the Agile Domain if OWSM is not installed. The configureWSSecurity tool is shipped under the Agile Installation folder and can be used to extend the Agile Domain to add the OWSM component.

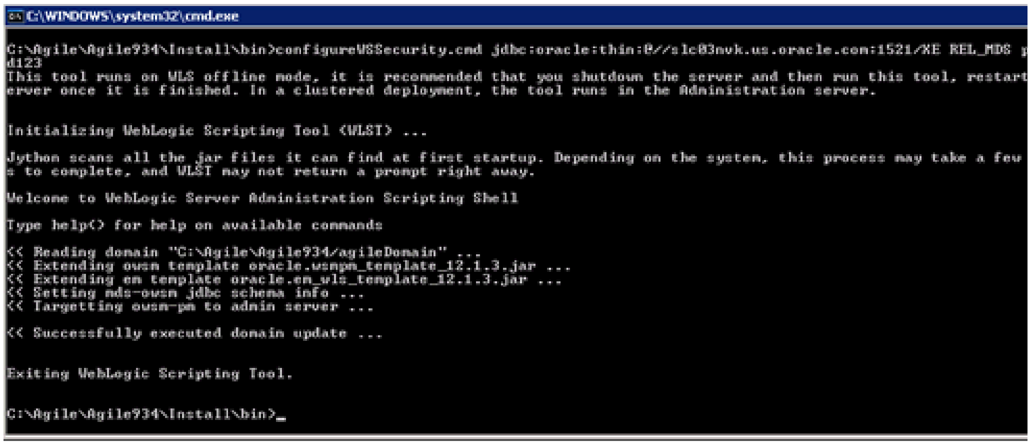
To enable the OWSM component in the Agile domain, do the following:

Note: You must back up the entire Agile Domain folder and corresponding RCU database before extending the domain.

1. Shut down the Agile server and back it up.
2. Open the Command Prompt and navigate to C:\Agile\Agile934\Install\bin.
3. Execute the following command:

   ```
   configureWSSecurity.cmd <RCU_DB_URL> <RCU_MDS_USER> <RCU_MDS_USER_
   PASSWORD>
   ```

Figure 7–1 Agile Domain extension for OWSM



```
C:\Agile\Agile934\Install\bin>configureWSSecurity.cmd jdbc:oracle:thin:@//slc03nvk.us.oracle.com:1521/XE REL_MDS p
d123
This tool runs on WLS offline mode, it is recommended that you shutdown the server and then run this tool, restart
erver once it is finished. In a clustered deployment, the tool runs in the Administration server.


Initializing WebLogic Scripting Tool (WLST) ...

Jython scans all the jar files it can find at first startup. Depending on the system, this process may take a few
s to complete, and WLST may not return a prompt right away.

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

<< Reading domain "C:\Agile\Agile934/agileDomain" ...
<< Extending owsm template oracle.wsmpm_template_12.1.3.jar ...
<< Extending em template oracle.em_wls_template_12.1.3.jar ...
<< Setting mds-owsm jdbc schema info ...
<< Targetting owsm-pm to admin server ...

<< Successfully executed domain update ...


Exiting WebLogic Scripting Tool.


C:\Agile\Agile934\Install\bin>_
```

Configuring WSS Policy for Agile PLM Web Services

Once the Agile Domain extension is completed, run the WSSConfigurator tool to secure A9 web services as described in the following steps.

> **Note:** Make sure that the Agile Server is running, and that the File Manager server is down.

1. Navigate to folder <agilehome>\agileDomain\tools, and unzip wssconfigurator.zip.
2. Run the WSSConfigurator tool and select option [1] to enable security.
    - **a.** In the wizard that follows, the user is prompted to provide the WLS server URL, username, and password to log in to the WebLogic Script Tool (WLST).
    - **b.** The tool automatically attaches an OWSM group policy to the A9 web services deployed on WebLogic. The wizard prompts you to choose the security configuration for the Reference Object web service client, which is used by A9 to call external web services. Choose the correct one according to the security policy defined on the Reference Object web service.
    - **c.** If you installed Agile Server and File Manager Server on the same server box, the WSSConfigurator tool automatically secures the File Manager web services, as well.
    - **d.** If you installed File Manager on another server, you need to do Step 2.a on that server. Then run the WSSConfigurator tool on the File Manager server, which automatically secures the File Manager web services for you.

      The wizard prompts you to provide the proper file.server.saml.privatekey.alias (SAML) and file.server.saml.token.issuer (Agile PLM). This updates the Server.conf file as below:

      ```
      # 934 WSS Service Configuration
      ```

```
file.server.saml.privatekey.alias=saml

file.server.saml.token.issuer=AgilePLM
```

Configuring WSS Policy for File Manager Web Services

In the previous section, the WSSConfigurator tool automatically created a key pair which is the Agile Server SAML signature key (its alias is AgileWssSamlSignKey) on the WebLogic server for SAML usage. The WSSConfigurator tool, however, can not automatically do the same for File Manager.

Generating File Manager SAML Signature Key

This section describes the steps to manually generate the File Manager SAML signature key.

1. Open the Command Prompt and navigate to C:\fm-keystore-934 folder.
2. Execute the following command to generate the File Manager SAML signature key into fm-keystore.jks.

   ```
   keytool -genkey -keyalg RSA -alias saml -keystore fm-keystore.jks
   -keysize 2048 -storepass agile123
   ```

   Enter the following information for the key generation command below: CN=FM SAML Signing, OU=Agile File Manager, O=Oracle, L=San Jose, S=CA, C=US
3. Export File Manager SAML Signing Key certificate by executing following command:

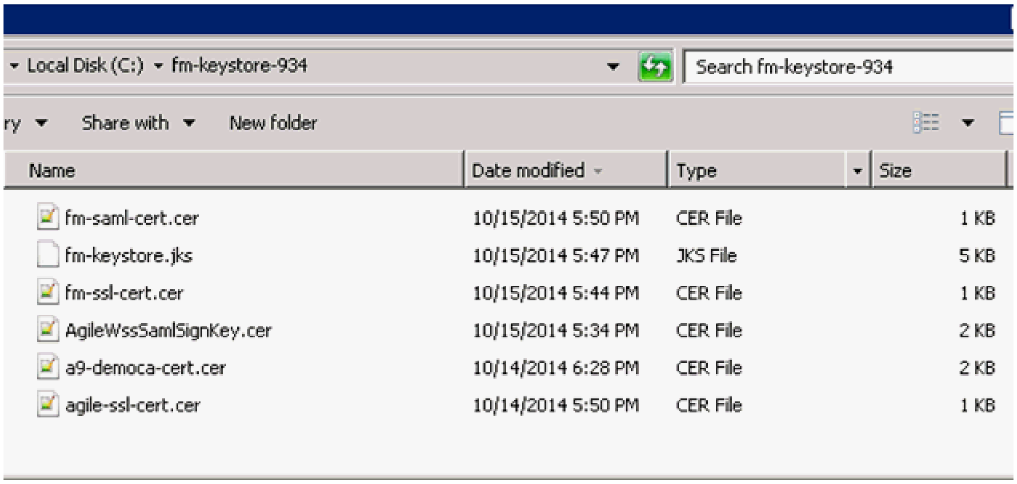
   ```
   keytool -export -file fm-saml-cert.cer -keystore fm-keystore.jks
   -storepass agile123 -alias saml
   ```

Import Agile Server SAML Signature Certificate into Tomcat Keystore

You need to import the Agile Server SAML signature certificate, and its trust chain, into the File Manager keystore. In this guide, we are using a self-signed certificate, which is signed by WLS Demo CA, so here we need to export WLS Demo CA and import it with the Agile Server SAML signature certificate in the order of Root to Leaf.

1. Launch the WSSConfigurator tool on the Agile Server and enter option [4] : To manage OPSS OWSM keystore and press enter. For the second prompt, enter option [5]: Export Demo CA certificate.

   The WSSConfigurator tool home folder contains two certificates, the Agile Server SAML Signature Certificate: AgileWssSamlSignKey.cer and the WLS Demo CA certificate: a9-democa-cert.cer.
2. Copy the two certificates to C:\fm-keystore-934. There should be more certificates in c:\fm-keystore-934 folder as shown.



Name	Date modified	Type	Size
fm-saml-cert.cer	10/15/2014 5:50 PM	CER File	1 KB
fm-keystore.jks	10/15/2014 5:47 PM	JKS File	5 KB
fm-ssl-cert.cer	10/15/2014 5:44 PM	CER File	1 KB
AgileWssSamlSignKey.cer	10/15/2014 5:34 PM	CER File	2 KB
a9-democa-cert.cer	10/14/2014 6:28 PM	CER File	2 KB
agile-ssl-cert.cer	10/14/2014 5:50 PM	CER File	1 KB

3. Import the WLS Demo CA certificate into the File Manager Keystore using the following command:

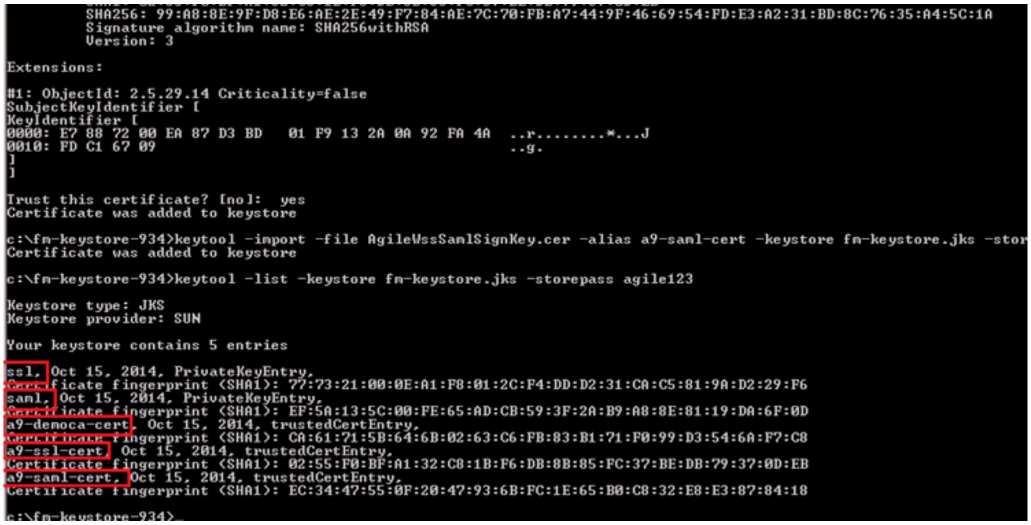
   ```
   keytool -import -file a9-democa-cert.cer -alias a9-democa-cert
   -keystore fm-keystore.jks -storepass agile123
   ```

4. Import the Agile Server SAML Signature certificate into File Manager Keystore using the following command:

   ```
   keytool -import -file AgileWssSamlSignKey.cer -alias a9-saml-cert
   -keystore fm-keystore.jks -storepass agile123
   ```

5. Verify the keys/certificates in File Manager Keystore using following command:

   ```
   keytool -list -keystore fm-keystore.jks -storepass agile123
   ```



```
         SHA256: 99:A8:8E:9F:D8:E6:AE:2E:49:F7:84:AE:7C:70:FB:A7:44:9F:46:69:54:FD:E3:A2:31:BD:8C:76:35:A4:5C:1A
         Signature algorithm name: SHA256withRSA
         Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: E7 88 72 00 EA 87 D3 BD   01 F9 13 2A 0A 92 FA 4A  ..r........*...J
0010: FD C1 67 09                                        ..g.
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore

c:\fm-keystore-934>keytool -import -file AgileWssSamlSignKey.cer -alias a9-saml-cert -keystore fm-keystore.jks -stor
Certificate was added to keystore

c:\fm-keystore-934>keytool -list -keystore fm-keystore.jks -storepass agile123

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 5 entries

ssl, Oct 15, 2014, PrivateKeyEntry,
Certificate fingerprint (SHA1): 77:73:21:00:0E:A1:F8:01:2C:F4:DD:D2:31:CA:C5:81:9A:D2:29:F6
saml, Oct 15, 2014, PrivateKeyEntry,
Certificate fingerprint (SHA1): EF:5A:13:5C:00:FE:65:AD:CB:59:3F:2A:B9:A8:8E:81:19:DA:6F:0D
a9-democa-cert, Oct 15, 2014, trustedCertEntry,
Certificate fingerprint (SHA1): CA:61:71:5B:64:6B:02:63:C6:FB:83:B1:71:F0:99:D3:54:6A:F7:C8
a9-ssl-cert, Oct 15, 2014, trustedCertEntry,
Certificate fingerprint (SHA1): 02:55:F0:BF:A1:32:C8:1B:F6:DB:8B:85:FC:37:BE:DB:79:37:0D:EB
a9-saml-cert, Oct 15, 2014, trustedCertEntry,
Certificate fingerprint (SHA1): EC:34:47:55:0F:20:47:93:6B:FC:1E:65:B0:C8:32:E8:E3:87:84:18

c:\fm-keystore-934>_
```

Import File Manager SAML Signature Certificate into Agile Server Keystore

By default the certificate exported by the keytool command is DER encoded binary X.509 certificate. You must convert it to Base64 encoded certificate.

1. Launch Command Prompt, navigate to the WLS domain directory and run setDomainEnv.cmd.

2. Navigate to the C:\fm-keystore-934 folder, and execute the following command to change the certificate suffix from.cer to .der.

   ```
   mv fm-saml-cert.cer fm-saml-cert.der
   ```

3. Execute the following command in the Command Prompt to convert the .der certificate to Base64 encoded .pem certificate.

   ```
   java utils.der2pem fm-saml-cert.der
   ```

   The fm-saml-cert.pem file is now in the C:\fm-keystore-934 folder.

4. Copy the fm-saml-cert.pem to C:\agile-keystore-934 folder.
5. Launch the WSSConfigurator tool on Agile Server to import File Manager SAML signature certificate. Select option [4]: To manage OPSS OWSM keystore.
6. Provide the WLS server URL, username and password as prompted. Once these WLS details are verified, select option [2]: Import Trusted Certificate.
7. Enter FMWssSamlSignKey as the alias name and enter C:\agile-keystore-934\fm-saml-cert.pem as the absolute certificate file path.
8. Choose option [1]: List All aliases. The aliases FMWssSamlSignKey and FMWssSamlSignKey are displayed.

Configure Trusted Issuer Using WSSConfigurator

For the SAML policy, OWSM looks for Issuer name, certificate DN, and compares them with the existing certificates in the OWSM keystore. This section describes how to configure the Trusted Issuer on both the Agile server and File Manager server.

Register Trusted SAML Issuer on Agile Server

To register a trusted SAML Issuer on the Agile Server, do the following:

1. Launch the WSSConfigurator on the Agile server, and select option [5]: To Add Token Issuer. Provide the WLS server URL, username, password as prompted to proceed.
2. Enter the Token Issuer Name and Token Issuer DN.

> **Note:** It is supported to have one DN for one Issuer Name. If you are using one SAML signature key everywhere then your system will have only one Issuer Name, or if you are using SAME DN in all SAML signature keys then you just need one Issuer Name. Otherwise, you need to configure this tool for several times to add each Issuer Name and Issuer DN pair.
>
> In this guide, we just have File Manager SAML signature key configured, so we just add one entry (Issuer Name: AgilePLM, Issuer DN: The full subject name of File Manager SAML signature key).

File Manager Application SAML Configuration

Open the file <AgileHomePath>\agileDomain\config\server.conf. You should see two lines similar as follows:

file.server.saml.privatekey.alias=saml

file.server.saml.token.issuer=AgilePLM

> **Note:** The private key alias should be the alias name you used when generating File Manager SAML signature key. In our guide, it is 'saml'
>
> The token issuer should be the Issuer Name you configured above. In this guide, it should be 'AgilePLM'.

Restart the WebLogic server and Tomcat server. Web Services Security is enabled for all A9 and File Manager web services.

Configuring WSS Policy For WSX

For WSX web services, policies are attached at design time by using the following annotation.

weblogic.wsee.jws.jaxws.owsm.SecurityPolicy

Below is an example of attaching security policy. Add the annotation to the implementation class of your web service.

Example 7–1

```
import weblogic.wsee.jws.jaxws.owsm.SecurityPolicy;

@WebService(portName = "SampleService", serviceName = "SampleService")
@SecurityPolicy(uri="oracle/wss11_saml_or_username_token_with_message_protection_
service_policy")
public class SampleServiceImpl implements SampleObjects {
```

Configuring WSS Policy for Reference Object Web Service

This section provides information on configuring Web Service Security for Reference Object web services.

Configure Server Policy for Reference Object WS

Agile PLM ships a reference implementation of Reference Object web service. In this section, we introduce how to configure Web Service Security policy for this reference implementation.

1. Ensure that SSL is correctly configured on the WLS server where the current Reference Object WS is running.
2. Ensure that OWSM is installed in the current WLS Domain.
3. Remove the HTTP Basic Authentication configuration from <agileDomain>\applications\application.ear\extension.war\WEB-INF.
   - Remove elements of <Security-constraint>,<Security-role>, and <Login-config> from web.xml.
   - Remove elements of <Security-role-assignment> from weblogic.xml.
4. Add WSS Annotation for Reference Object WS Reference Implementation.
   a. Modify <referenceswsx>\build.xml to add the following line in the build.classpath element.

      <pathelement path="<Oracle_Home>/oracle_common/modules/clients/com.oracle.webservices.fmw.client_12.1.3.jar "/>

b. Add the following Security Policy annotation in the Java file: <referenceswsx>\src\com\agile\integration\externalreference\services\service\v1\ QuickViewReferenceObjectImpl.java.

   import weblogic.wsee.jws.jaxws.owsm.SecurityPolicy;

   @SecurityPolicy(uri="oracle/wss11_saml_or_username_token_with_message_protection_service_policy")

c. Add following Security Policy annotation in Java file: <referenceswsx>\src\com\agile\integration\externalreference\services\service\v1\ SearchReferenceObjectsImpl.java.

   import weblogic.wsee.jws.jaxws.owsm.SecurityPolicy;

   @SecurityPolicy(uri="oracle/wss11_saml_or_username_token_with_message_protection_service_policy")

d. Run Ant to compile this reference implementation and deploy it.

5. Restart the Agile Server, and then this Reference Object WS will be protected by WSS policy.

Configure Client Policy for Reference Object WS Client

1. Open Java Client, go to Settings > System Settings > Reference Objects Management, and change the Host Base URL to use https, change port to 7002.
2. Run the WSSConfigurator tool, and choose the Client Policy for Reference Object WS client as introduced in "Configuring WSS Policy for Agile PLM Web Services" on page 7-2.
3. If you choose SAML token client policy, copy the AgileWssSamlSignKey.cer file under WSSConfigurator tool home directory to Reference Object WS service server. Then run the WSSConfigurator tool on the server side, choose option [4], then option [2] to import the client side AgileWssSamlSignKey to the server side, using alias name 'AgileWssSamlSignCert'.
4. Import the Reference Object WS server SSL certificate to Client WLS trust key store.
5. Restart the Agile Server. The Reference Object WS client should be able to invoke the secure web services.

WSS Policy Map

After enabling the web service security, the attached service side policy and certified corresponding client policies are listed in the table below:

Table 7–1

Web Services	Service Policy	Client Policy
Core Service	oracle/wss11_saml_or_username_token_with_message_protection_service_policy	oracle/wss_username_token_over_ssl_client_policy oracle/wss_saml_token_bearer_over_ssl_client_policy oracle/wss11/saml_token_with_message_protection_client_policy
EC Service AIS Service Reference Object	oracle/wss11_saml_or_username_token_with_message_protection_service_policy	oracle/wss_username_token_over_ssl_client_policy oracle/wss_saml_token_bearer_over_ssl_client_policy

8

Disabling Security

We recommend that you have security enabled in your Agile PLM environment, but if you choose to disable it, refer to the following sections.

Disabling SSL

To disable SSL, you must do the following:

1. Turn off the SSL flag in WebLogic server.
2. Remove the lines in the startup script in the Tomcat server that were introduced for SSL configurations, as in section "Configuring SSL on the Tomcat Server" on page 6-10.

Disabling Web Services Security

To disable Web Services Security:

1. Launch the WSSConfigurator tool on the Agile Server.
2. To disable web service security, enter option [2].

> **Note:** File Manager policy is also disabled in the same process.

3. Choose the option to disable Reference Object client policy.

A

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.

   a. Use a firewall.

   b. Never poke a hole through a firewall.

   c. Monitor who accesses your systems.

   d. Check network IP addresses.

   e. Encrypt network traffic.

   f. 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.

B

Checklist for Configuring Web Services Security

The checklists in this section list the main tasks needed to configure Web Services security. It includes configuration steps for SSL setup described in Chapter 6, "Configuring SSL" and WSS setup that is described in Chapter 7, "Enabling Security for Web Services".

A9 and File Manager Web Services Setup Checklist

Table B–1 Checklist for A9 and FM Web Services Security Setup Tasks

Task	Complete?	Comments
Create A9 Keystore(jks) and SSL key		
Export A9 SSL certificate		
Create A9 trust store and import A9 SSL certificate		
Enable SSL port in Weblogic console		
Configure keystore and trust store in Weblogic console		
Select SSL key as SSL Private key in Weblogic console		
Enable Security using wssconfugrator		
Export Demo CA certificate using wssconfigurator		
Create FM Keystore and SSL key		
Export FM SSL certificate		
Import A9 SSL certificate in FM keystore		
Import FM SSL certificate in A9 keystore		
Change Server.xml to enable SSL		
Change Server.conf to use https protocol and SSL ports		
Change Java Client Web Server URL to HTTPS and change Port to SSL		

Table B–1 (Cont.) Checklist for A9 and FM Web Services Security Setup Tasks

Task	Complete?	Comments
Import AgileWssSignKey into FM keystore		
Import a9-democa-cert into FM Keystore		
Generate FM SAML key		
Export FM SAML cert		
Create Base64 of FM SAML certificate		
Import Base64 FM SAML cert into A9 using wssconfigurator		
Create Trust Issuer using wssconfigurator		

Distributed File Manager Configuration Checklist

Table B–2 Checklist for DFM Configuration Tasks

Task	Complete?	Comments
Create DFM Keystore and SSL key		
Export DFM SSL certificate		
Change DFM server.conf with appropriate URLs		
Import A9 and all FM SSL certificates (All FM, DFM1, DFM2, IFS, and so on)		
Import all FM SAML keys into DFM keystore (if SAML key is different in all FMs)		
Import DFM SSL certificates into A9 trust store and All FMs ke		
Create Trust issuer if different SAML key is used		
Update Issuer Name in server.conf		

Autovue Configuration Checklist

Table B–3 Checklist for Autovue Configuration Tasks

Task	Complete?	Comments
Import all SSL certs (FM,DFM, SSL, DemoCA) into Autovue/jre/security/cacerts		

C

SSL Security Configurations for Developers

This chapter discusses information that is 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.

SDK Client Configuration

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 functionality of the Agile Application Server. Using the SDK, you can create programs that extend the functionality of Agile PLM and can perform tasks against the PLM system.

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

Configuring SSL for SDK

To configure SSL for SDK, do the following:

1. Get the certification key, for example mykeystore.jks, that is generated using the steps in Appendix B, and keep the mykeystore.jks file in a folder located on the system 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 the file 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. Execute run.bat.

Web Service Client Configuration

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

Web Service Extensions

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 web services using SSL. For more information about how to configure SSL for web services, refer to Chapter 7, "Enabling Security for Web Services".

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

D

WS Security Configurations for Developers

This appendix provides information regarding Web Service Security configurations for developers.

Configuring WSS for Web Service Client

The following sections provide some examples of using the sample code.

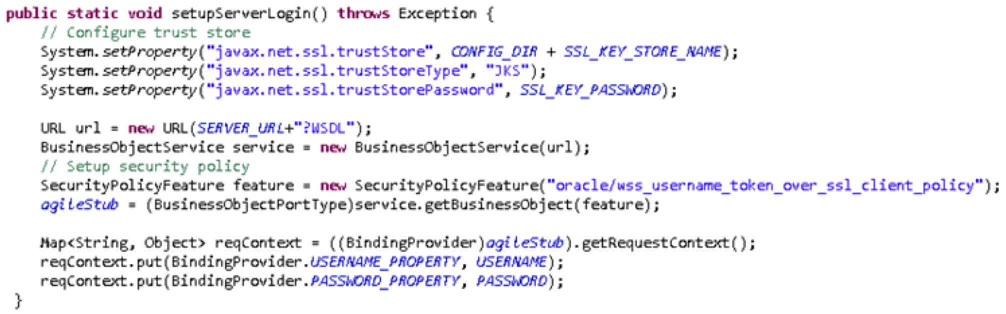
For more information about Agile web services, see *Agile Web Services User Guide*.

Using Username Token Over SSL Policy

You need to configure the SSL certificate. Get the certificate, for example, sslclient.crt, and use the following command to generate a keystore sslclient.jks and import the certificate.

```
keytool -import -keystore sslclient.jks -storepass password -alias
sslclientkey -file sslclient.crt
```

Configure the sample code as below and change all the required binding properties according to your environment. Make sure that the required jar file, com.oracle.webservices.fmw.client_12.1.3.jar, which is under Weblogic_HOME\oracle_common\modules\clients\, is added.



```
public static void setupServerLogin() throws Exception {
    // Configure trust store
    System.setProperty("javax.net.ssl.trustStore", CONFIG_DIR + SSL_KEY_STORE_NAME);
    System.setProperty("javax.net.ssl.trustStoreType", "JKS");
    System.setProperty("javax.net.ssl.trustStorePassword", SSL_KEY_PASSWORD);

    URL url = new URL(SERVER_URL+"?WSDL");
    BusinessObjectService service = new BusinessObjectService(url);
    // Setup security policy
    SecurityPolicyFeature feature = new SecurityPolicyFeature("oracle/wss_username_token_over_ssl_client_policy");
    agileStub = (BusinessObjectPortType)service.getBusinessObject(feature);

    Map<String, Object> reqContext = ((BindingProvider)agileStub).getRequestContext();
    reqContext.put(BindingProvider.USERNAME_PROPERTY, USERNAME);
    reqContext.put(BindingProvider.PASSWORD_PROPERTY, PASSWORD);
}
```

In the example, the settings were as follows:

`SSL_KEY_STORE_NAME = sslclient.jks`

SSL_KEY_PASSWORD = password

Using SAML Token Bearer Policy

In order to use the SAML token bearer policy on the client side, you need to configure a signed key pair and SSL certificate. Complete the following steps to do the configuration.

Generate a SAML Signature Key

1. Generate a signed key and import it into the server OPSS keystore in the server. The following command generates a signed key pair to be stored in JseSignKeyStore.jks:

   ```
   keytool -genkeypair -alias JseSignKey -keystore JseSignKeyStore.jks
   -keyalg RSA -sigalg SHA1withRSA -validity 3650 -dname
   cn=Test,ou=Agile,O=Oracle,L=Test,ST=Test,C=Test -storepass password
   -keypass password
   ```

2. Use the -list option to check if the key is successfully generated. The alias name is jsesignkey

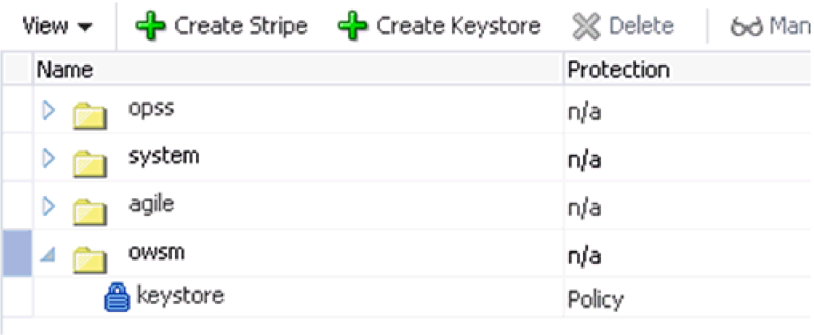
   ```
   keytool -list -keystore JseSignKeyStore.jks -storepass password
   ```

3. Export the public key.

   ```
   keytool -exportcert -keystore JseSignKeyStore.jks -alias jsesignkey
   -storepass password -rfc
   ```

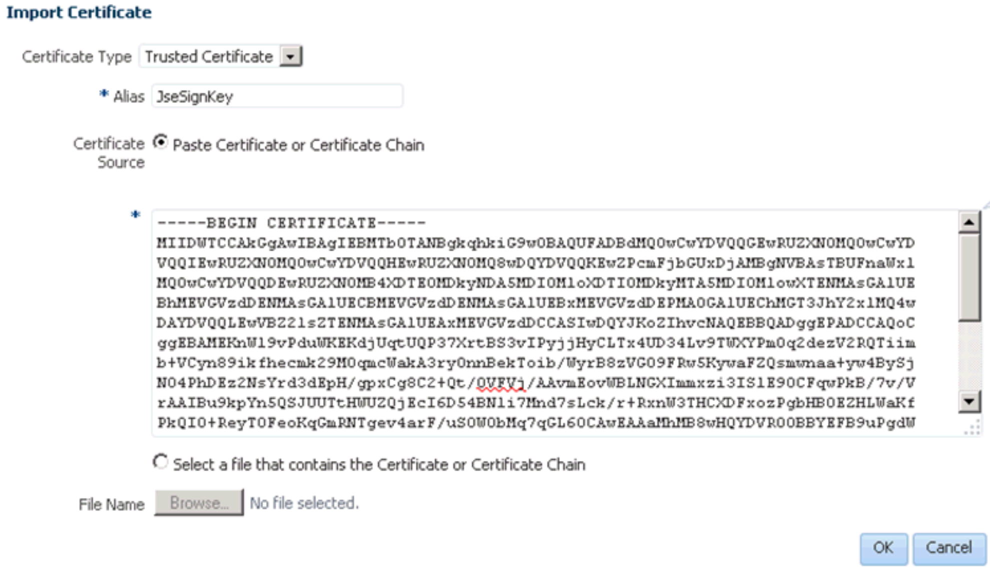
4. Import the public key to the server. In Oracle Enterprise Manager on the Agile server, open agileDomain > WebLogic Domain menu > Security > Keystore.

   In the Keystore configuration UI, check if the owsm row exists. If not, create a row named "owsm" and a keystore named "keystore",



Name	Protection
opss	n/a
system	n/a
agile	n/a
owsm	n/a
keystore	Policy

   Select keystore under owsm and click Manage > Import. This opens the Import Certificate page for you to import the public key. Follow the example below and paste the encoded key context exported in step 3. Note the alias name.



Import Certificate

Certificate Type: Trusted Certificate

* Alias: JseSignKey

Certificate Source: ● Paste Certificate or Certificate Chain

*
```
-----BEGIN CERTIFICATE-----
MIIDWTCCAkGgAwIBAgIEBMTb0TANBgkqhkiG9w0BAQUFADBdMQ0wCwYDVQQGEwRUZXN0MQ0wCwYD
VQQIEwRUZXN0MQ0wCwYDVQQHEwRUZXN0MQ8wDQYDVQQKEwZPcmFjbGUxDjAMBgNVBAsTBUFnaWxl
MQ0wCwYDVQQDEwRUZXN0MB4XDTE0MDkyNDA5MDI0M1oXDTI0MDkyMTA5MDI0MlowXTENMAsGA1UE
BhMEVGVzdDENMAsGA1UECBMEVGVzdDENMAsGA1UEBxMEVGVzdDEPMA0GA1UEChMGT3JhY2xlMQ4w
DAYDVQQLEwVBZ2lsZTENMAsGA1UEAxMEVGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAMEKnW19vPduWKEKdjUqtUQP37XrtBS3vIPyjjHyCLTx4UD34Lv9TWXYPm0q2dezV2RQTiim
b+VCyn89ikfhecmk29M0qmcWakA3ry0nnBekToib/WyrB8zVG09FRw5KywaFZQsmwnaa+yw4BySj
N04PhDEz2NsYrd3dEpH/gpxCg8C2+Qt/0VFVj/AAvmEovWBLNGXImmxzi3IS1E90CFqwPkB/7v/V
rAAIBu9kpYn5QSJUUTtHWUZQjEcI6D54BN1i7Mnd7sLck/r+RxnW3THCXDFxozPgbHB0EZHLWaKf
PkQI0+ReyT0FeoKqGmRNTgev4arF/uS0W0bMq7qGL60CAwEAAaMhMB8wHQYDVR00BBYEFB9uPgdW
```

○ Select a file that contains the Certificate or Certificate Chain

File Name: Browse... No file selected.

If all is successful, there should be a certificate under owsm/keystore named as JseSignKey.

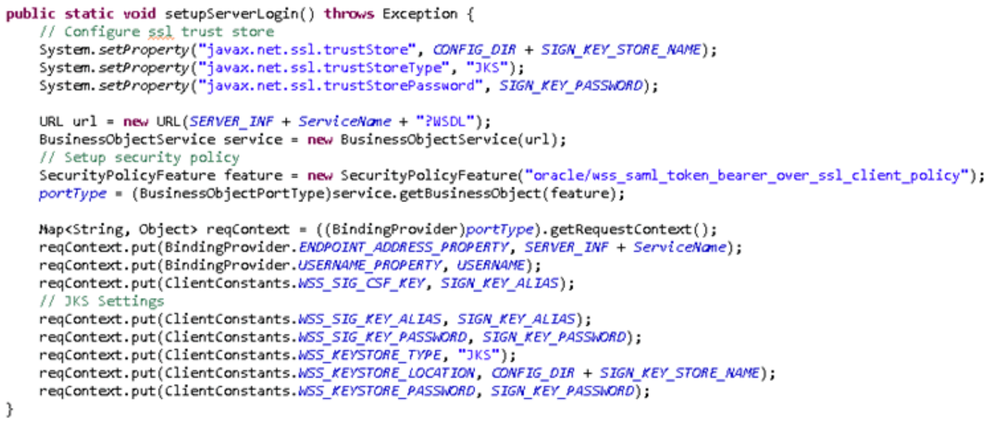
Configure SSL Certificate

Get the certificate, for example, sslclient.crt, and use the following command to import it to the keystore JseSignKeyStore.jks generated in step 1 in "Generate a SAML Signature Key" or generate a new jks.

```
keytool -import -keystore JseSignKeyStore.jks -storepass password -alias
sslclientkey -file sslclient.crt
```

Configure Sample Code

Change all of the required binding properties according to your environment. Make sure the required jar com.oracle.webservices.fmw.client_12.1.3.jar, which is under Weblogic_HOME\oracle_common\modules\clients\, is added.



```
public static void setupServerLogin() throws Exception {
    // Configure ssl trust store
    System.setProperty("javax.net.ssl.trustStore", CONFIG_DIR + SIGN_KEY_STORE_NAME);
    System.setProperty("javax.net.ssl.trustStoreType", "JKS");
    System.setProperty("javax.net.ssl.trustStorePassword", SIGN_KEY_PASSWORD);

    URL url = new URL(SERVER_INF + ServiceName + "?WSDL");
    BusinessObjectService service = new BusinessObjectService(url);
    // Setup security policy
    SecurityPolicyFeature feature = new SecurityPolicyFeature("oracle/wss_saml_token_bearer_over_ssl_client_policy");
    portType = (BusinessObjectPortType)service.getBusinessObject(feature);

    Map<String, Object> reqContext = ((BindingProvider)portType).getRequestContext();
    reqContext.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, SERVER_INF + ServiceName);
    reqContext.put(BindingProvider.USERNAME_PROPERTY, USERNAME);
    reqContext.put(ClientConstants.WSS_SIG_CSF_KEY, SIGN_KEY_ALIAS);
    // JKS Settings
    reqContext.put(ClientConstants.WSS_SIG_KEY_ALIAS, SIGN_KEY_ALIAS);
    reqContext.put(ClientConstants.WSS_SIG_KEY_PASSWORD, SIGN_KEY_PASSWORD);
    reqContext.put(ClientConstants.WSS_KEYSTORE_TYPE, "JKS");
    reqContext.put(ClientConstants.WSS_KEYSTORE_LOCATION, CONFIG_DIR + SIGN_KEY_STORE_NAME);
    reqContext.put(ClientConstants.WSS_KEYSTORE_PASSWORD, SIGN_KEY_PASSWORD);
}
```

In this example, the settings were as follows:

```
SIGN_KEY_ALIAS = JseSignKey

SIGN_KEY_PASSWORD = password

SIGN_KEY_STORE_NAME = JseSignKeyStore.jks
```

Execute the sample. If everything is configured properly, the sample should work with the web service secured with SAML token bearer policy.

E

SSL Protocol and Signature Algorithm Changes

The section describes SSL protocol and signature algorithm changes.

Signature Algorithm Changes

SHA-1 is not good enough for security purposes, so CA/Browser Forum voted to deprecate SHA-1 certificates entirely for SSL/TLS connections with the termination of issuing any new certificates. Agile PLM is certified on SHA-2.

Deselecting SSL 3.0

Follow these steps to deselect SSL 3.0.

Server Client Settings

Complete the following procedures, as appropriate, to deselect SSL 3.0 from the server client.

Excluding SSL 3.0 on Oracle WebLogic Server 12c

Add the following system property to the setUserOverrides file located in the <Agile_HOME>\agileDomain\bin folder:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Dweblogic.security.SSL.protocolVersion=TLS1
```

The property for Unix: export JAVA_OPTIONS="$JAVA_OPTIONS -Dweblogic.security.SSL.protocolVersion=TLS1"

Oracle WebLogic Server 12c uses JSSE as the default SSL implementation. This setting enables any protocol that starts with "TLS".

Excluding SSL 3.0 on Tomcat V7

Add the sslEnabledProtocols setting and remove the sslProtocol="TLS" in the server.xml file:

```
<Connector protocol="org.apache.coyote.http11.Http11Protocol"

port="8443" maxThreads="200"

scheme="https" secure="true" SSLEnabled="true"

keystoreFile="C:\fm-keystore-935\fm-keystore.jsk"
```

keystorePass="agile123" keyAlias="ssl"

clientAuth="false" sslEnabledProtocols="TLSv1, TLSv1.1, TLSv1.2"/>

Excluding SSL 3.0 on WSS Configuration Tool Before Enabling WSS

Extract wssconfigurator.zip to a folder named wssconfigurator under <Agile_HOME>\agileDomain\tools\.

Add the following system property to the wssconfigurator.sh file located in <Agile_HOME>\agileDomain\tools\wssconfigurator\:

WLST_PROPERTIES="-Dweblogic.security.SSL.protocolVersion=TLS1"

export WLST_PROPERTIES

User Client Settings

Complete the following procedures, as appropriate, to deselect SSL 3.0 from the user client side.

Disabling SSL 3.0 for Applets and Webstarts

If you have multiple JREs installed, you must identify which JRE is used locally, then navigate to its installation folder and run %JRE_HOME%\bin\javacpl.exe to start the Java Control Panel:

1. Click the Advanced tab.
2. Under Advanced Security Settings, deselect Use SSL 3.0.
3. Click Apply.
4. Click the Java tab.
5. Click View... to view the Java Runtime settings.
6. Add -Dweblogic.security.SSL.protocolVersion=TLS1 to the Runtime Parameters field for the selected JRE.
7. Click OK.

> **Note:** Changes made to the Control Panel while the browser is open take effect only after the browser is restarted. Java WebStart applications, like the Agile PLM Java Client, must also be restarted for changes to take effect.

Disabling SSL 3.0 for Java Applications

Table E–1 System Properties to Disable SSL 3.0

JDK Version	System Property to Disable SSL 3.0
JDK 5, 6, 7	java -Dhttps.protocols="TLSv1" -Dweblogic.security.SSL.protocolVersion=TLS1 <MyApp>
JDK 8 and above	Java -Dweblogic.security.SSL.protocolVersion-TLS1 -Dhttps.protocols="TLSv1, TLSv1.1, TLSv1.2" -Djdk.tls.client.protocols="TLSv1, TLSv1.1,TLSv1.2" <MyApp>

Disabling SSL 3.0 for Browsers

Table E–2 Steps to Disable SSL 3.0 By Browser

Browser	Steps to disable SSL 3.0
Internet Explorer	1. On the Internet Explorer Tools menu, click Internet Options. 2. In the Internet Options dialog box, click the Advanced tab. 3. In the Security category, uncheck Use SSL 3.0 and make sure the following are checked: Use TLS 1.0, Use TLS 1.1, and Use TLS 1.2 (if available). Note: It is important to check consecutive versions. Not selecting consecutive versions could result in connection errors. 4. Click OK. 5. Exit and restart Internet Explorer.
Mozilla Firefox	1. Type about:config in the Firefox address bar and press Enter. 2. Click I’ll be careful, I promise! 3. Type security.tls.version in the search bar. 4. Double-click the preference of "security.tls.version.min" and set its value to 1. 5. Restart Firefox. Alternatively, you can install the Firefox Extension SSL Version Control which provides a graphical way to specify the minimum SSL version.
Chrome	Chrome does not have a configurable setting in the user interface to turn off SSL 3.0. Instead, Chrome needs to be told not to use SSL 3.0 at launch. To automatically launch Chrome with SSL 3.0 disabled, run Chrome with the command Chrome.exe -ssl-version-min=tls1 to specify that the minimum version of SSL to be used is TLs 1.0.
Safari	There is no setting for Safari to disable SSL 3.0. You must upgrade the Safari browser to the latest version. Apple has released Security Update 2014-005 which disables CBC-mode ciphers in coordination with SSL 3.0.