

Agile Product Lifecycle Management

Security Guide

Release 9.3.6

E71146-07

April 2019

Agile Product Lifecycle Management Security Guide, Release 9.3.6

E71146-07

Copyright © 2013, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

Contributing Author: Edlyn Sammanasu

Contributor:

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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	vii
 1 Document Scope	
Documentation Audience	1-1
Guide to this Document	1-1
 2 Agile PLM Overview	
 3 Overview of Security Fundamentals	
Basic Security Considerations	3-1
Keep Software Up-To-Date	3-1
Restrict Network Access to Critical Services	3-1
Follow the Principle of Least Privilege	3-1
Monitor System Activity	3-2
Keep Up To Date on Latest Security Information	3-2
 4 Performing a Secure Agile PLM Installation	
Understanding the Agile PLM Environment	4-1
Recommended Deployment Topologies	4-1
Determining Installation Flows Based on Security Needs	4-3
Agile Setup Without SSL or WSS Enabled	4-4
Agile Setup With Only SSL Enabled	4-4
Agile Setup With SSL and WSS Enabled	4-5
Prerequisites Before Installing Agile PLM	4-7
Installing the Oracle Database Server	4-7
Installing Oracle Fusion Middleware 12.2.1.1	4-7
Installing Agile PLM	4-7
Optional Component Configuration	4-8
Configuring AutoVue (Optional)	4-8
Configuring MCAD Connectors (Optional)	4-8

5 Protecting Agile PLM Data

Password Policy	5-1
Configuring and Using Authentication	5-1
LDAP-based Authentication	5-1
SSO-based Authentication	5-2
Database-based Authentication	5-3
Configuring and Using Access Control	5-3
Configuring and Using Security Audit	5-3
User Monitor	5-3
History Tab	5-4
Log Files	5-4

6 Configuring SSL

Securing Agile PLM Application Using SSL	6-2
Generating WebLogic SSL Signature Key and Certificate Signing Request	6-2
Importing CA Certificate To WebLogic SSL Keystore	6-2
Generating WebLogic SSL Truststore	6-2
Configuring SSL on WebLogic Server	6-2
Configuring the Keystore on the Weblogic Server	6-3
Configuring the Identity of the WebLogic Server	6-3
Configuring SSL Listen Port for WebLogic Server	6-3
Verify SSL Configuration on WebLogic Server	6-3
Cluster Environment: Additional Configurations	6-3
Configuring SSL in the Agile PLM Application Server	6-3
HTTPOnly and SecureFlag Flags in agile.properties	6-4
Securing Agile PLM File Manager(s) Using SSL	6-5
Generating SSL Signature Key and Certificate Signing Request for File Manager	6-5
Importing CA Certificate To File Manager SSL Keystore	6-5
Configuring SSL on the File Manager	6-5
Offloading SSL To OHS	6-6
Installing OHS	6-6
Configuring SSL On OHS Server	6-6
Replacing Demo Certificate With CA Signed Certificate	6-6
Generating a New Wallet	6-8
Generating New Key Pair	6-8
Generating Certificate Signing Request (CSR)	6-8
Import CA Signed Certificate, Root CA and Intermediate CA Certificate	6-8
Import Key from JKS to Wallet	6-8
Change Path of Wallet	6-9
Configuring OHS	6-9
Configuring Virtual Host	6-9
Configuring OHS WebLogic Module	6-9
Configuring OHS Reverse Proxy for File Manager	6-10
Enable Weblogic Plugin	6-11
Changing Configuration on the Agile PLM Application Server	6-11
Changing Server URLs In JavaClient	6-12
Changing Configuration On The File Manager Server	6-12

Changing Server URLs In server.conf.....	6-13
Generate File Manager Trusted Keystore.....	6-13
Updating File Manager server.xml	6-13
Configuring SSL on AutoVue Server.....	6-13
Configuring SSL on Distributed File Managers(DFMs)	6-14
Configuring SSL on AutoVue Client	6-14
7 Enabling Security for Web Services	
Installing OWSM on the Agile Domain	7-2
Configuring WSS Policy for Agile PLM Web Services	7-2
Configuring Agile Server SAML Signature Key	7-3
Configuring WSS Policy for File Manager Web Services	7-4
Generating File Manager SAML Signature Key and Certificate Signing Request	7-4
Import Agile Server SAML Signature Certificate into File Manager Keystore.....	7-4
Import File Manager SAML Signature Certificate into Agile Server Keystore.....	7-4
Configure Trusted Issuer Using WSSConfigurator	7-5
Register Trusted SAML Issuer on Agile Server.....	7-5
File Manager Application SAML Configuration.....	7-5
WSS Policy Map	7-6
Configuring WSS Policy For WSX	7-6
Configuring WSS Policy for Reference Object Web Service	7-6
Configure Server Policy for Reference Object WS.....	7-7
Configure Client Policy for Reference Object WS Client.....	7-7
8 Disabling Security	
Disabling SSL	8-1
Disabling Web Services Security	8-1
A Secure Deployment Checklist	
B Checklist for Configuring Web Services Security	
A9 and File Manager Web Services Setup Checklist	B-1
Distributed File Manager Configuration Checklist	B-2
Autovue Configuration Checklist.....	B-3
C SSL Security Configurations for Developers	
SDK Client Configuration.....	C-1
Configuring SSL for SDK	C-1
Web Service Client Configuration.....	C-2
Web Service Extensions	C-2
D WS Security Configurations for Developers	
Configuring WSS for Web Service Client.....	D-1
Using Username Token Over SSL Policy	D-1
Using SAML Token Bearer Policy	D-2

Generate a SAML Signature Key	D-2
Configure SSL Certificate.....	D-2
Configure Sample Code	D-2

E SSL Protocol and Signature Algorithm Changes

Signature Algorithm Changes	E-1
Deselecting SSL 3.0	E-1
Server Client Settings.....	E-1
Excluding SSL 3.0 on Oracle WebLogic Server 12c.....	E-1
Excluding SSL 3.0 on Tomcat 8.5	E-1
Excluding SSL 3.0 on WSS Configuration Tool Before Enabling WSS.....	E-2
User Client Settings.....	E-2
Disabling SSL 3.0 for Applets and Webstarts	E-2
Disabling SSL 3.0 for Java Applications	E-2
Disabling SSL 3.0 for Browsers	E-3

F Setting up AutoVue 21.0.1 for HTTPS

Internet Explorer/Google Chrome	F-1
Mozilla Firefox.....	F-1
Using CCK2 to Import a Certificate	F-1
Using Certutil to Import a Certificate	F-2
Safari.....	F-3

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.
<i>italic</i>	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.

Document Scope

This document provides IT users and Agile PLM administrators with the information needed to 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.

Agile PLM Overview

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

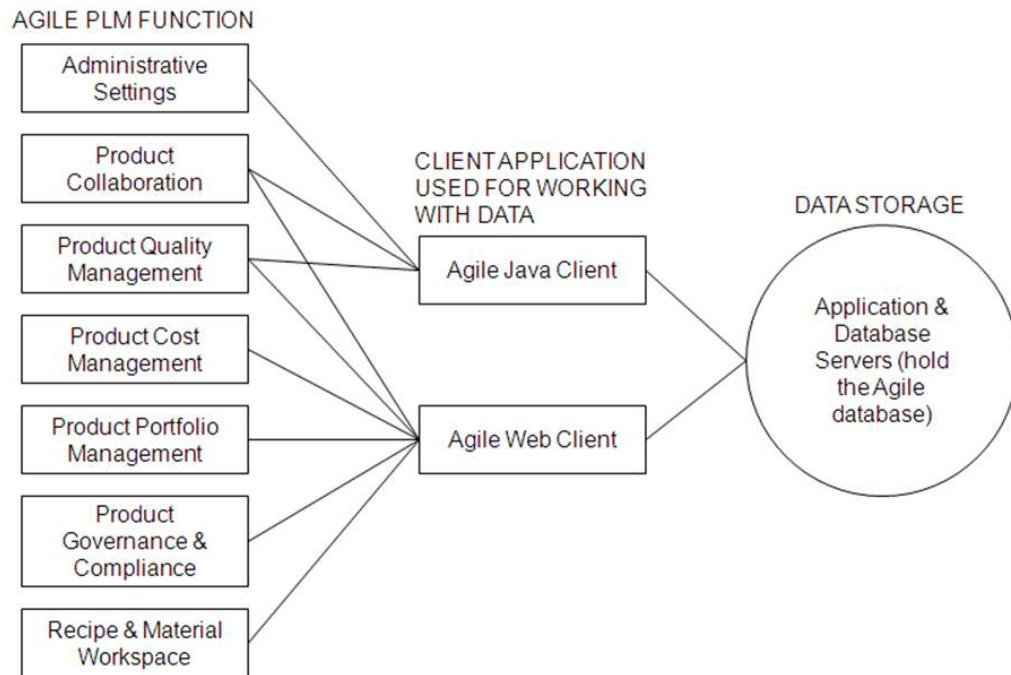
- **Agile Product Collaboration (PC)** - Management and collaboration of product record information throughout the product lifecycle, across internal organizations and the extended supply chain. Accessed through Web Client and Java Client.
- **Agile Product Governance & Compliance (PG&C)** - Management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



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.

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?**

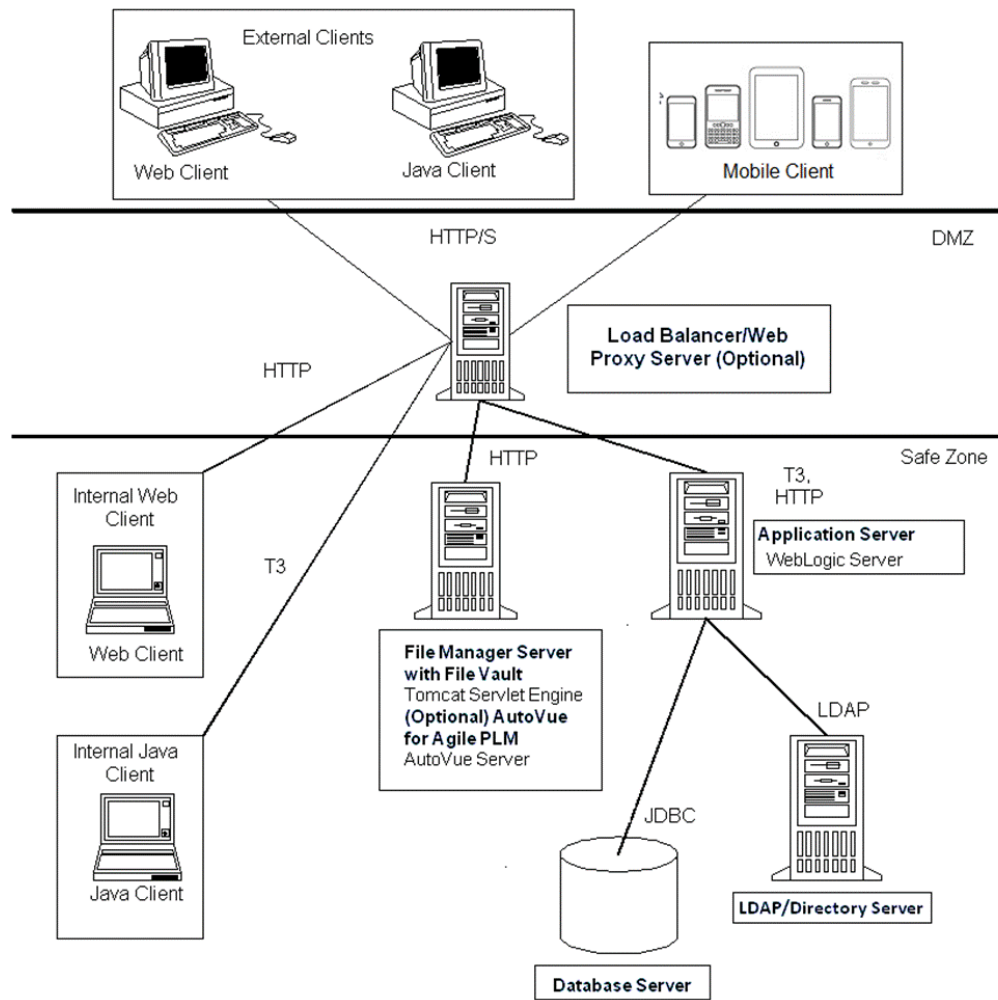
For example, you must protect your subscribers' data from other subscribers, but someone in your organization might need to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, perhaps a system administrator can manage your system components 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 "[Configuring SSL](#)" on page 6-1 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:

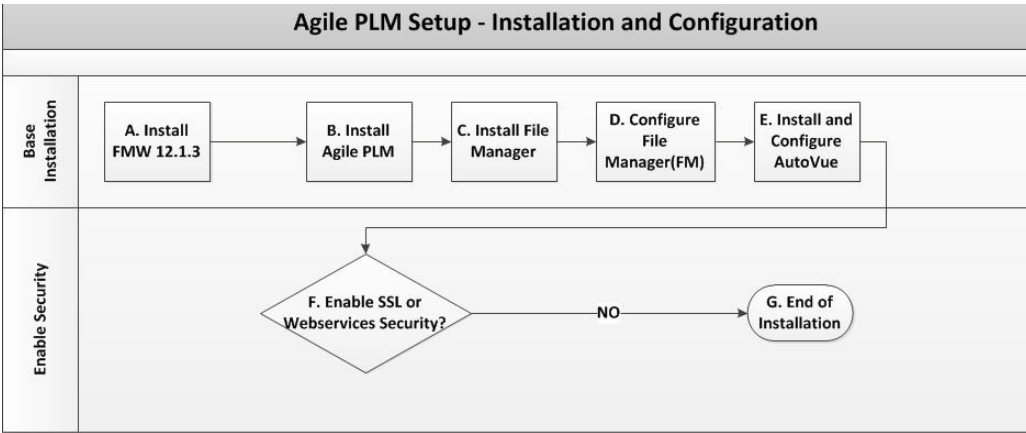
1. Agile installation without SSL or WSS enabled. See "[Agile Setup Without SSL or WSS Enabled](#)" on page 4-4
2. Agile installation with only SSL. See "[Agile Setup With Only SSL Enabled](#)" on page 4-4.

- 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.

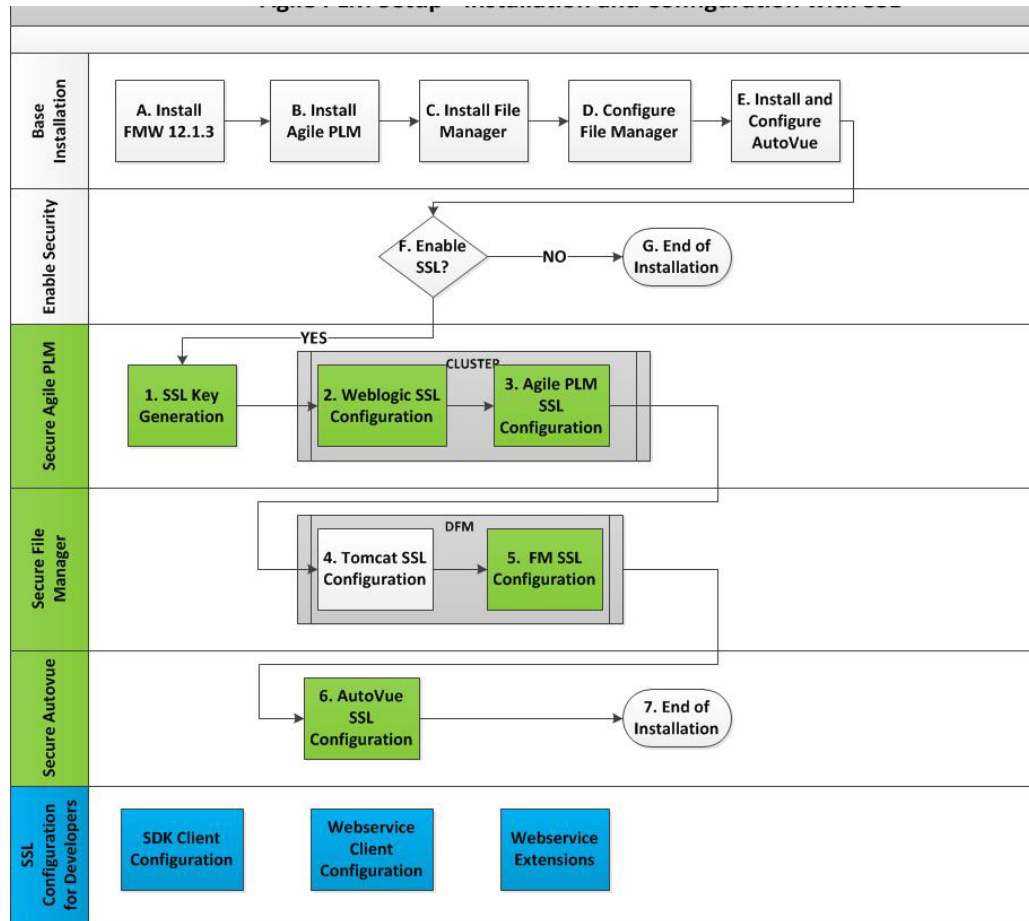
Figure 4–2 Agile PLM Setup Without SSL or WSS



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.

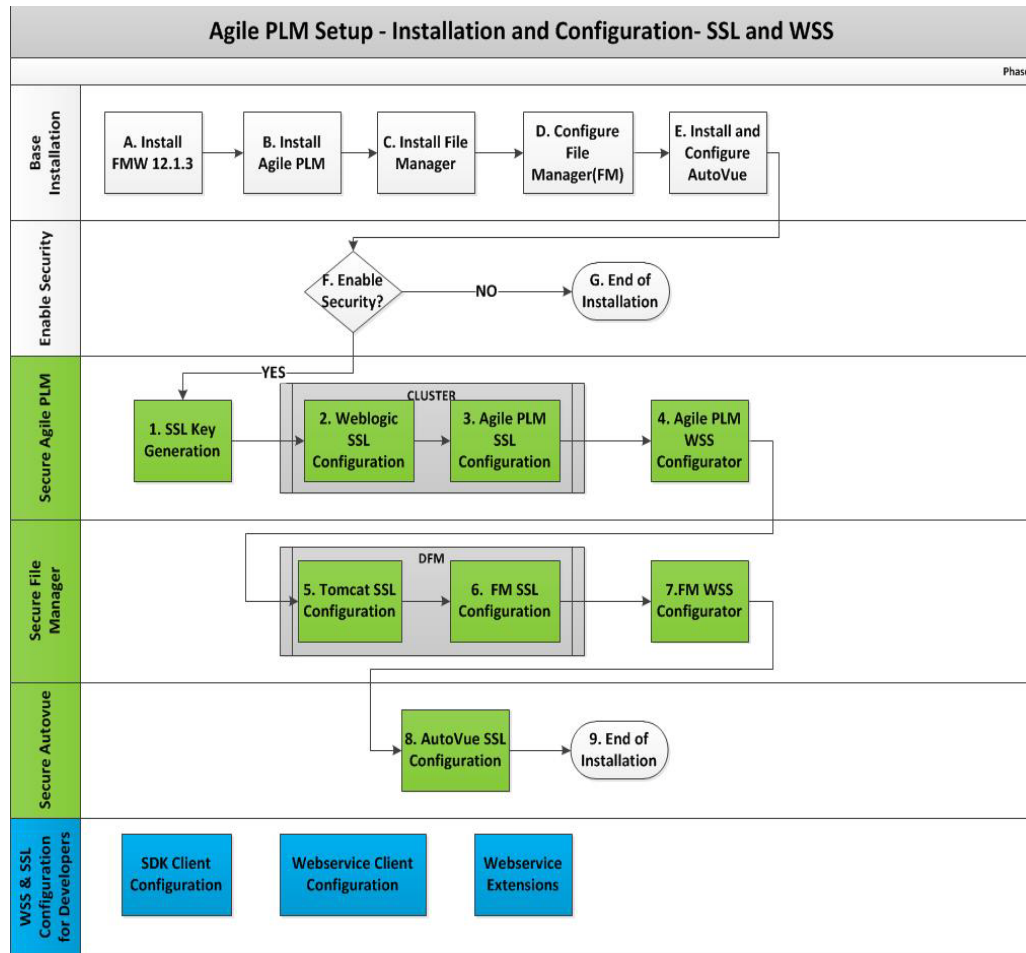
Figure 4–3 Agile PLM Setup with Only SSL

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 ["Configuring SSL"](#) on page 6-1, especially ["Securing Agile PLM Application Using SSL"](#), ["Configuring SSL on WebLogic Server"](#) and ["Configuring SSL in the Agile PLM Application Server"](#).
3. To complete steps 4-5, follow the relevant procedures ["Configuring SSL"](#) on page 6-1, especially ["Securing Agile PLM File Manager\(s\) Using SSL"](#).
4. To complete step 6, follow the relevant procedures [Chapter 6, "Configuring SSL"](#), especially ["Enable Weblogic Plugin"](#).
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

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 ["Securing Agile PLM Application Using SSL"](#), ["Configuring SSL on WebLogic Server"](#) and ["Configuring SSL in the Agile PLM Application Server"](#).
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 ["Enable Weblogic Plugin"](#).

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 12.2.1.1

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

Optional Component Configuration

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

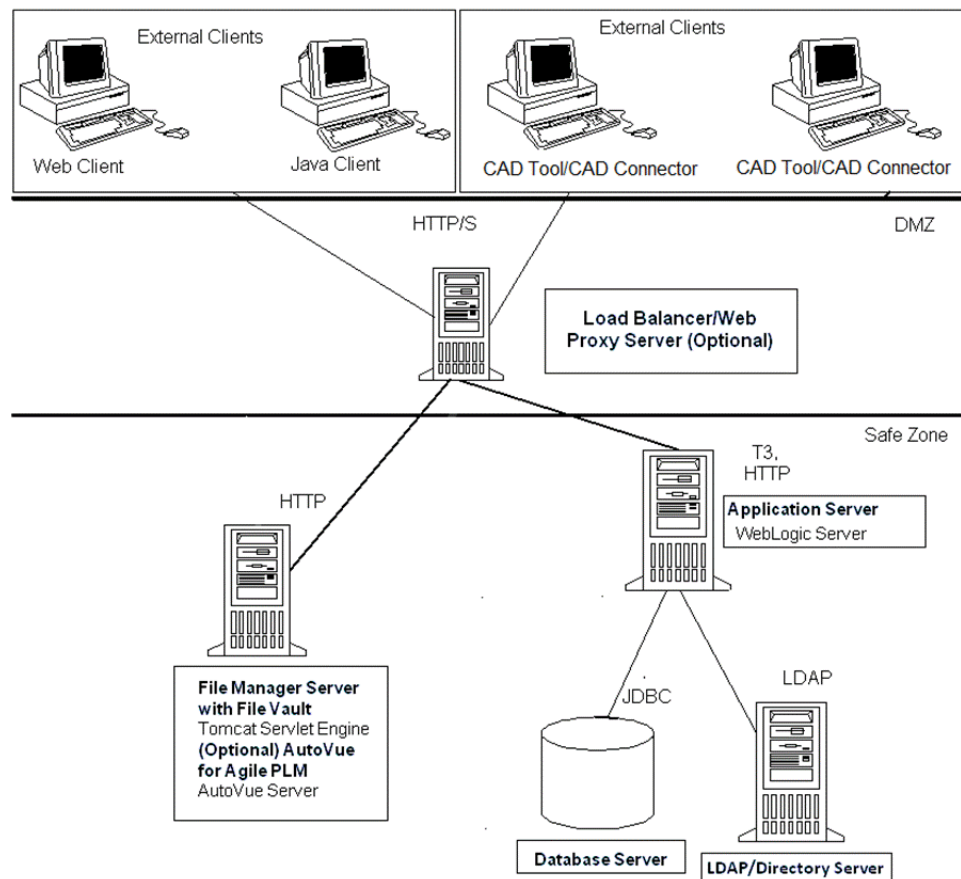
Configuring AutoVue (Optional)

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.

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 can be integrated with various Single Sign-On (SSO) solutions. SSO is a web-based authentication solution and thus can be enabled for and used in the web client only.

Single Sign-On enables centralized security management including management of user credentials and access policies and can reduce the frequency and circumstances that the user is challenged for their credentials. When Agile PLM is configured for Single Sign-On, a user that has already been authenticated with the system (for example, with the operating system or a corporate portal) may not be prompted again for their credentials by the application.

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

- SAML 2.0 SSO (SAML2) with Oracle IDCS as Identity Provider - may also work with other identity providers
- Oracle Access Manager (OAM)
- Microsoft Windows NT LAN Manager (NTLM)

For details on configuring Agile PLM with the above SSO solutions, refer to Appendix A of the *Agile PLM Administrator Guide* entitled "Configuring Single Sign-On".

Note: With some SSO solutions, PX/SDK applications, web service calls and AgileDrive (aka WebDAV) may not be able to connect to an Agile PLM application URL protected by SSO. In this case, calls must be made to an alternative entry point that is not protected by SSO.

URL PX Authentication

Customers can use Process Extensions (PXs) to extend the Agile PLM user interface or business logic. Agile PLM provides a process for the PX to access the Agile PLM server without having to explicitly provide credentials by passing a one-time-use encrypted authentication token to the application. The token is secured by storing it in the Agile PLM Database so that it is not accessible outside the application. This token can be used to ensure that a given instance of the PX call is valid and is executed only once after the PX has been executed by the user. Once the validation has been successful, the token is deleted before providing Agile PLM server access to the PX. Finally, the token itself expires after a certain time interval and this expiration time is configurable. Oracle recommends that this interval be configured to minimize potential misuse.

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*.

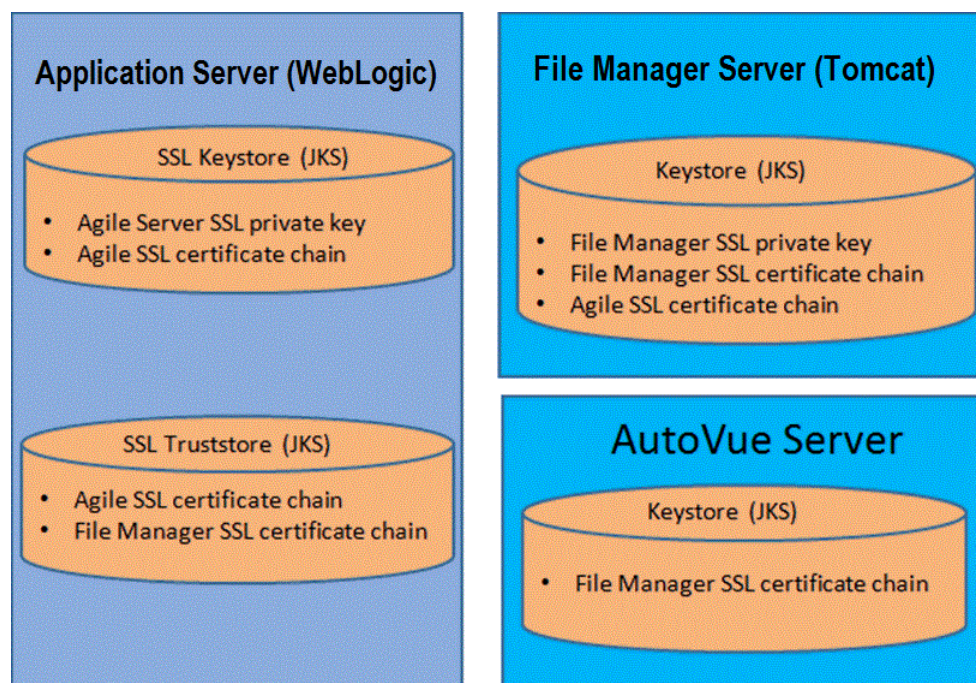
For more information about enabling logging scripts in WebLogic, see "Application Logging and WebLogic Logging Services" in the WebLogic Server documentation.

Configuring SSL

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

The following diagram introduces the required keystores/keys for SSL configurations.

Figure 6–1 Required Keystores/keys for SSL



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

- SDK
- Web Services
- Application Server (WebLogic)
- File Manager Server (Tomcat)
- 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.

Note: For instructions on how to mitigate vulnerabilities related to SSL 3.0 and SHA-2 certificate, 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.

To set up SSL, you need three keystores. In this document, they will be named as follows:

- Agile Server SSL keystore: agile-keystore.jks
- Agile Server SSL truststore: agile-truststore.jks
- File Manager SSL keystore: fm-keystore.jks

Securing Agile PLM Application Using SSL

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

Generating WebLogic SSL Signature Key and Certificate Signing Request

To generate the WebLogic SSL Signature Key and Certificate Signing Request, do the following:

1. Generate SSL keystore, agile-keystore.jks.
 - Alias: ssl
 - Key size: 2048
 - Algorithm: RSA
2. Generate Certificate Signing Request with the SSL keystore above and send to the Certifying Authority.

Importing CA Certificate To WebLogic SSL Keystore

The Certifying Authority returns the newly issued certificate, the Root CA and an intermediate CA certificate. Importing 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) The Root CA (our trust anchor CA) and of (b) intermediate SSL CA before (c) your newly issued SSL certificate is installed.

Generating WebLogic SSL Truststore

Generate the WebLogic SSL truststore, agile-truststore.jks. To generate this truststore, import your Root CA, Intermediate SSL CA, and Issued CA certificates into the keystore, agile-truststore.jks, that constitutes the trust.

Configuring SSL on WebLogic Server

Once you have imported the CA certificate to WebLogic SSL keystore and generated the WebLogic SSL truststore, 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.
4. In AgileServer > Configuration > Keystores, use Custom Identity and Custom Trust for keystores.
 - In the Identity Section provide the following:
 - Enter the location in the Custom Identity Keystore field.
 - Enter "JKS" as the Custom Identity Keystore Type.
 - Enter the password in the Custom Identity Keystore Passphrase field.
 - In the Trust Section provide the following:
 - Enter the location in the Custom Trust Keystore field.
 - Enter "JKS" as the Custom Keystore Type.
 - Enter the password in the Custom Trust Keystore Passphrase
 - Click Save.

Configuring the Identity of the WebLogic Server

Go to AgileServer > Configuration > SSL. In this example, we use "ssl" is the key, and the password.

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

Verify SSL Configuration on WebLogic Server

1. Connect to https://<hostname>:7002/Agile/PLMServlet and confirm that you can access Agile Web Client successfully.
2. 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 Server

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: <AGILE_HOME>\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">

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 Tomcat Server configuration file (AGILE_HOME\FileManager\conf\server.xml) 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.

Generating SSL Signature Key and Certificate Signing Request for File Manager

To generate the SSL signature key and Certificate Signing Request for File Manager, do the following:

1. Generate SSL keystore fm-keystore.jks.
 - Alias: fm
 - Key size: 2048
 - Algorithm: RSA
2. Generate the Certificate Signing Request with the SSL keystore above and send it to the Certifying Authority.

Importing CA Certificate To File Manager SSL Keystore

The Certifying Authority returns the newly issued certificate, the Root CA and an intermediate CA certificate. Importing 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) The Root CA (our trust anchor CA) and of (b) intermediate SSL CA before (c) your newly issued SSL certificate is installed.

Configuring SSL on the File Manager

Once you have imported the CA certificate to the File Manager SSL keystore, continue with the following procedures.

1. Export the File Manager SSL certificate from fm-keystore.jks, which we named as fm-ssl-cert.cer. Import File Manager SSL certificate into Agile Server SSL Trust Keystore.
2. Export Agile Server certificate from agile-keystore.jks, which we named as agile-ssl-cert.cer. Import agile-ssl-cert.cer into File Manager key store.
3. Open <AGILE_HOME>\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.Http11NioProtocol"port="8443"
maxThreads="200"
```

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

```
keystoreFile="<certificate path>\fm-keystore.jks" keystorePass=<keystore_
password> keyAlias="fm" clientAuth="false" sslProtocol="TLS"/>
```

4. To configure SSL on the File Manager application, change <AGILE_HOME>\agileDomain\config\server.conf as follows:

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
5. 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
6. 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.

Offloading SSL To OHS

You can offload SSL from the Agile PLM application server and File Manager server to a web proxy server or hardware load balancer, which will simplify the SSL configuration. Agile PLM is certified to have SSL terminated at OHS(Oracle HTTP Server), all communication traffic from a client to OHS is over HTTPS protocol, and over HTTP between OHS and Agile PLM application server and File Manager server.

Installing OHS

Down load the Oracle HTTP Server 12.2.1.1 from OSDC, then follow the guide <https://docs.oracle.com/middleware/1221/core/install-ohs/GUID-D5AFD830-8A7D-42CC-8C22-CE68C452CF4A.htm#WTINS-GUID-D5AFD830-8A7D-42CC-8C22-CE68C452CF4A> to install and configure OHS in a standalone domain. Please pay attention to Missing Libraries Might Cause HTTPD to Exit Without Notice if you are installing OHS on Windows OS.

Configuring SSL On OHS Server

Use the following to configure SSL on an OHS Server:

Replacing Demo Certificate With CA Signed Certificate

After you have completed the configuration in a standalone domain, you can start the OHS instance. By default, SSL is enabled on port 4443 with a demo certificate. The private-public key pair is stored in demo wallet configured in <OHS_DOMAIN>/config/fmwconfig/components/OHS/<OHS_INSTANCE>\ssl.conf

The Listen directive below has a comment preceding it that is used


```

# by tooling which updates the configuration. Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen 4443
##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost *:4443>
  <IfModule ssl_module>
    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on
    # Client Authentication (Type):
    # Client certificate verification type and depth. Types are
    # none, optional and require.
    SSLVerifyClient optional
    # SSL Protocol Support:
    # Configure usable SSL/TLS protocol versions.
    SSLProtocol ALL
    # SSL Cipher Suite:
    # List the ciphers that the client is permitted to negotiate.
    SSLCipherSuite TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_
    ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_
    256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_
    ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_
    128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_RC4_128_SHA,TLS_ECDHE_ECDSA_
    WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_GCM_
    SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_
    WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_
    SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_
    AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_RC4_128_SHA,TLS_ECDHE_RSA_
    WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_
    WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_
    WITH_AES_128_CBC_SHA256,SSL_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_
    WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA,SSL_RSA_WITH_3DES_
    EDE_CBC_SHA
    # SSL Certificate Revocation List Check
    # Valid values are On and Off
    SSLCRLCheck Off
    #Path to the wallet

```

```
SSLWallet "${ORACLE_
INSTANCE}/config/fmwconfig/components/${COMPONENT_
TYPE}/instances/${COMPONENT_NAME}/keystores/default"

</IfModule>

</VirtualHost>
```

Generating a New Wallet

Open a command console and execute the following command to create an Auto-Login Wallets:

```
<Oracle_Home>\oracle_common\bin\orapki wallet create -wallet new_wallet -auto_
login_only
```

If you want to create a password protected wallet, please refer to this MOS.

Doc ID 1629906.1

Generating New Key Pair

```
<Oracle_Home>\oracle_common\bin\orapki wallet add -wallet new_wallet -dn
"CN=hostname.us.oracle.com,OU=TESTING PURPOSE ONLY,O=Oracle
Corporation,L=Redwood City,ST=California,C=US" -asym_alg RSA -sign_alg sha256
-keysize 2048 -auto_login_only
```

Generating Certificate Signing Request (CSR)

Generate a CSR and send it to Certifying Authority.

```
<Oracle_Home>\oracle_common\bin\orapki wallet export -wallet new_wallet -dn
"CN=hostname.us.oracle.com, OU=TESTING PURPOSE ONLY,O=Oracle Corporation,L=Redwood
City, ST=California,C=US" -request myserver.csr
```

Import CA Signed Certificate, Root CA and Intermediate CA Certificate

Once you get the CA signed certificate along with the Root CA and Intermediate Ca certificate, import the Root CA and Intermediate Ca certificate into the wallet as trusted certificate, and import your server certificate as user certificate.

```
<Oracle_Home>\oracle_common\bin\orapki wallet add -wallet new_wallet -trusted_cert
-cert VTN-PublicPrimary-G5.pem -auto_login_only
```

```
<Oracle_Home>\oracle_common\bin\orapki wallet add -wallet new_wallet -trusted_cert
-cert Oracle_SSL_CA_G2.pem -auto_login_only
```

```
<Oracle_Home>\oracle_common\bin\orapki wallet add -wallet new_wallet -user_cert
-cert cert.cer -auto_login_only
```

To display the content in the wallet you can run following command:

```
<Oracle_Home>\oracle_common\bin\orapki wallet display -wallet new_wallet
```

Import Key from JKS to Wallet

Use the following steps if you have an existing JKS keystore and want to import the key pair into a newly created wallet:

```
# Generate a PKCS12 file from JKS keystore using JDK keytool
keytool -importkeystore -srckeystore agile-keystore.jks -srcstorepass srcstorepass
-destkeystore sslkeystore.p12 -deststoretype PKCS12 -deststorepass deststorepass
-destkeypass keypass -srcalias ssl
```

```
# Import the PKCS12 into a wallet
```

```
<Oracle_Home>/oracle_common/bin/orapki wallet create -wallet new_wallet -auto_login_only
<Oracle_Home>/oracle_common/bin/orapki wallet import_pkcs12 -wallet new_wallet -auto_login_only -pkcs12file sslkeystore.p12 -pkcs12pwd agile123
```

Change Path of Wallet

Update the configuration file

<OHS_DOMAIN>/config/fmwconfig/components/OHS/<OHS_INSTANCE>\ssl.conf by editing below line:

```
SSLWallet
"${ORACLE_INSTANCE}/config/fmwconfig/components/${COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/new_wallet
```

After you have changed the wallet location in ssl.conf, restart the OHS instance. It will now use the CA signed certificate.

Configuring OHS

Use the following information to Configure OHS:

Configuring Virtual Host

Optionally create a virtual host to listen on the default HTTP port and redirect all requests to SSL port, or do not configure any HTTP listen port.

```
#[Listen] OHS_HTTP_PORT
Listen 7777
<VirtualHost *:7777>
    Redirect permanent / https://ohs_host:ssl_port/
</VirtualHost>
```

Configuring OHS WebLogic Module

Open the <OHS_DOMAIN>/config/fmwconfig/components/OHS/<OHS_INSTANCE>\mod_wl_ohs.conf

```
LoadModule weblogic_module "${PRODUCT_HOME}/modules/mod_wl_ohs.so"
```

This empty block is needed to save mod_wl related configuration from EM to this file when changes are made at the Base Virtual Host Level

```
<IfModule weblogic_module>
    Debug ON
    WLLogFile c:/temp/weblogic.log
    WLProxySSL ON
    <Location /Agile>
        WLSRequest on
        WebLogicHost weblogic_host
        WebLogicPort port
    #For Clustered environment, you can configure the WebLogicCluster as follow
    #WebLogicCluster host1:8002,host2:8003
</Location>
```

```
<Location /CoreService>
    WLSRequest on
    WebLogicHost weblogic_host
    WebLogicPort port
</Location>
<Location /JavaClient>
    WLSRequest on
    WebLogicHost weblogic_host
    WebLogicPort port
</Location>
</IfModule>
<VirtualHost *:4443>
```

Configuring OHS Reverse Proxy for File Manager

Modify the <OHS_DOMAIN>/config/fmwconfig/components/OHS/<OHS_INSTANCE>\ssl.conf by adding following directives into the SSL virtual host.

```
<IfModule ssl_module>
    ProxyPreserveHost On
    ProxyPass          /Filemgr http://fm_host:fm_port/Filemgr
    ProxyPassReverse   /Filemgr http://fm_host:fm_port/Filemgr
</IfModule>
```

If you want to offload SSL for DFM, you can configure another virtual host in the same OHS instance or in a separate OHS instance. If you configure the virtual host in the same OHS instance, you need to open up another listen port and enable SSL for it.

Listen 4445

```
<VirtualHost *:4445>
    <IfModule ssl_module>
        # SSL Engine Switch:
        # Enable/Disable SSL for this virtual host.
        SSLEngine on
        # Client Authentication (Type):
        # Client certificate verification type and depth. Types are
        # none, optional and require.
        SSLVerifyClient optional
        # SSL Protocol Support:
        # Configure usable SSL/TLS protocol versions.
        SSLProtocol ALL
        # SSL Cipher Suite:
        # List the ciphers that the client is permitted to negotiate.
```

```

    SSLCipherSuite TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_
    ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_
    256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_
    ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_
    128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_RC4_128_SHA,TLS_ECDHE_ECDSA_
    WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_GCM_
    SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_
    WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_
    SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_
    AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_RC4_128_SHA,TLS_ECDHE_RSA_
    WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_
    WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_
    WITH_AES_128_CBC_SHA256,SSL_RSA_WITH_AES_256_CBC_SHA,SSL_RSA_
    WITH_AES_128_CBC_SHA,SSL_RSA_WITH_RC4_128_SHA,SSL_RSA_WITH_3DES_
    EDE_CBC_SHA

    # SSL Certificate Revocation List Check

    # Valid values are On and Off

    SSLCRLCheck Off

    #Path to the wallet

    SSLWallet "${ORACLE_
    INSTANCE}/config/fmwconfig/components/${COMPONENT_
    TYPE}/instances/${COMPONENT_NAME}/keystores/default"

    ProxyPreserveHost On

    ProxyPass      /Filemgr http://DFM_HOST:DFM_PORT/Filemgr
    ProxyPassReverse /Filemgr http://DFM_HOST:DFM_PORT/Filemgr

</IfModule>

</VirtualHost>

```

Enable Weblogic Plugin

Enable the Weblogic Plug-In at domain level:

1. Login WLS Console
2. Navigate to Domain > Configuration > Web Applications
3. Check the checkbox with the text 'WebLogic Plug-In Enabled'
4. Save the changes and restart WebLogic Servers

Changing Configuration on the Agile PLM Application Server

Modify the following configuration files:

1. <AGILE_HOME>/agileDomain/config/agile.properties


```

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

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

WebSecurity.ForceSecureCookies = true

```
2. <AGILE_HOME>/agileDomain/application/application.ear/JavaClient.war/wls/ext.jnlp

- ```
<jnlp spec="1.0+" codebase="https://ohs_host:ohs_ssl_port/JavaClient">
```
3. `<AGILE_HOME>/agileDomain/application/application.ear/JavaClient.war/pcclient.jnlp`  

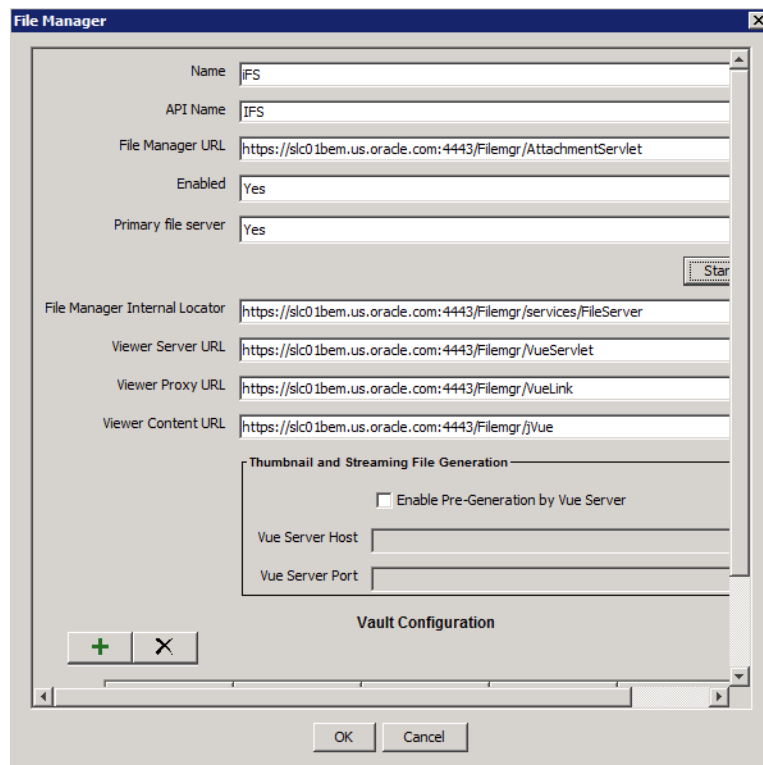
```
<jnlp spec="1.0+" codebase="https://ohs_host:ohs_ssl_port/JavaClient">
<argument>serverURL=t3://app_server:app_port</argument>
<argument>jvuecodebase=https://ohs_host:ohs_ssl_port/Filemgr/jVue</argument>
<argument>jvueserver=https://ohs_host:ohs_ssl_port/Filemgr/VueServlet</argument>
```
  4. `<AGILE_HOME>/agileDomain/application/application.ear/JavaClient.war/custom.jnlp`  

```
<jnlp spec="1.0+" codebase="https://ohs_host:ohs_ssl_port/JavaClient">
```

### Changing Server URLs In JavaClient

Use the following information to change the server URLs in JavaClient:

1. Login Java Client, go to Admin? Server Settings, open Locations.
2. Update the Web Server URL: `https://ohs_host:ohs_ssl_port/Agile/PLMServlet`.
3. Click Save.
4. Click the File Manager tab, update URLs for all File Managers



### Changing Configuration On The File Manager Server

User the following information to change the configuration on the File Manager Server:

### Changing Server URLs In server.conf

Modify the following property settings in <AGILE\_HOME>/agileDomain/config/server.conf

```
app.server.url =https://ohs_host:ohs_ssl_
port/Agile/FSHelper/FSHelperWSService
file.server.url =https://ohs_host:ohs_ssl_port/Filemgr/services/FileServer
dms.server.url =https://ohs_host:ohs_ssl_
port/Agile/DmsService/DmsViewerAPIService
```

### Generate File Manager Trusted Keystore

Generate trusted JKS keystore by JDK tool and import OHS SSL Certificates into File Manager trusted keystore

```
keytool -import -keystore fmTrustedkeystore.jks -storepass storepass -file
root.pem -alias rootca
keytool -import -keystore fmTrustedkeystore.jks -storepass storepass -file
intermediate.pem -alias intermediateca
keytool -import -keystore fmTrustedkeystore.jks -storepass storepass -file
cert.cer -alias ohsssl
```

### Updating File Manager server.xml

<AGILE\_HOME>/FileManager/conf/server.xml

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
truststoreFile="C:/Release/fmTrustedkeystore.jks" truststorePass="storepass"
proxyName="ohs_host" proxyPort="ohs_port"/>
```

If you enabled WSS with SSL terminated at OHS, please update the server.xml as below. Please note that the alias orakey referenced in the keyAlias parameter is the File Manager Server SAML signature key.

```
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000"
scheme="https" secure="true" keystoreFile="C:/Release/fmkeystore.jks"
keystorePass="storepass" keyAlias="orakey"
truststoreFile="C:/Release/fmtruststore.jks" truststorePass="truststorePass"
proxyName="ohs_server" proxyPort="ohs_port"/>
```

After changing the configuration and updating all the server URL settings in Java Client, restart the application servers to effect the changes.

## Configuring SSL on AutoVue Server

AutoVue server should be configured to point to SSL protected VueServlet which is hosted on File Manager.

1. Import both the Application server and File Manager server certificates into the AutoVue Server's JRE (<AGILE\_HOME>\jre\lib\security\cacerts) using Java's keytool command:

---

**Note:** The certificates have already been generated in steps 1 and 2 of ["Configuring SSL on the File Manager"](#) on page 6-5.

---

2. Restart the AutoVue server.

## Configuring SSL on Distributed File Managers(DFMs)

If there are multiple DFM nodes deployed, you need to do the following configurations on each node.

- Set up DFMs.
- Follow the steps on section Securing Agile PLM File Manager(s) Using SSL.
- Export DFMs SSL certificate.
- Import DFMs SSL certificate into Agile Server trust store (agile-truststore.jks) and File Manager keystore (fm-keystore.jks).
- Restart the file manager server and access `https://<fm_server_alias>:8443/Filemgr/Configuration` to check the File Manager configuration.

## Configuring SSL on AutoVue Client

---

---

**WARNING: This is for AutoVue 21.0.1 only**

---

---

In the Applet-Free AutoVue Client, the user's web browser sends commands to the client using the XMLHttpRequest API. If the user launches AutoVue from a secure page, the Mixed Active Content rules that are enforced by browsers requires that the browser also communicate to the AutoVue Client through an HTTPS channel. This requires that the AutoVue Client provide a server certificate. The certificate can be either self-signed, or signed with an existing local certificate authority.

To support this configuration, installers for products embedding AutoVue should follow these steps:

1. Generate a security certificate for "localhost". This certificate will only be used to enable SSL communication between the user's browsers and the AutoVue Client, so it should be as restricted as possible. An administrative tool is provided with AutoVue (<AutoVue Install Dir>\tools\makeAvCert) which produces suitable certificates. When the utility is run, it generates two files: `av_cert.pem`, which contains the complete certificate with keypair, and `localhost.cer`, which contains the public information.
2. Extract <Agile\_Home>\agileDomain\applications\application.ear by using tool `ExtractArchive(<Agile_Home>\Install\bin)`.
3. Copy the complete certificate file, `av_cert.pem`, to <Agile\_Home>\agileDomain\applications\ExpandedEar\JavaClient.war in the Admin server where it can be accessed by web browsers when their users have been authenticated.
4. Add the URL which will reference the SSL certificate file as a parameter to `autovue.jnlp` (<Agile\_Home>\agileDomain\applications\ExpandedEar\application.war\jVue):

```
<application-desc main-class="com.cimmetry.jvue.JVueApp">
<argument>-paramsslcert_url=https://<Agile_or_Proxy_Host>/JavaClient/av_
cert.pem
</argument>
...
```



</application-desc>

5. Repack <Agile\_Home>\agileDomain\applications\ExpandedEar by using tool RepackArchive(<Agile\_Home>\Install\bin).

---

**Note:** The use of self-signed localhost certificates requires that they are loaded as a certificate exception in each user's browser. The localhost.cer file generated by makeAvCert is intended for this purpose. For the best experience, this should be done before users attempt to use AutoVue. See [Appendix E, "SSL Protocol and Signature Algorithm Changes"](#) for details.

---



## Enabling Security for Web Services

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

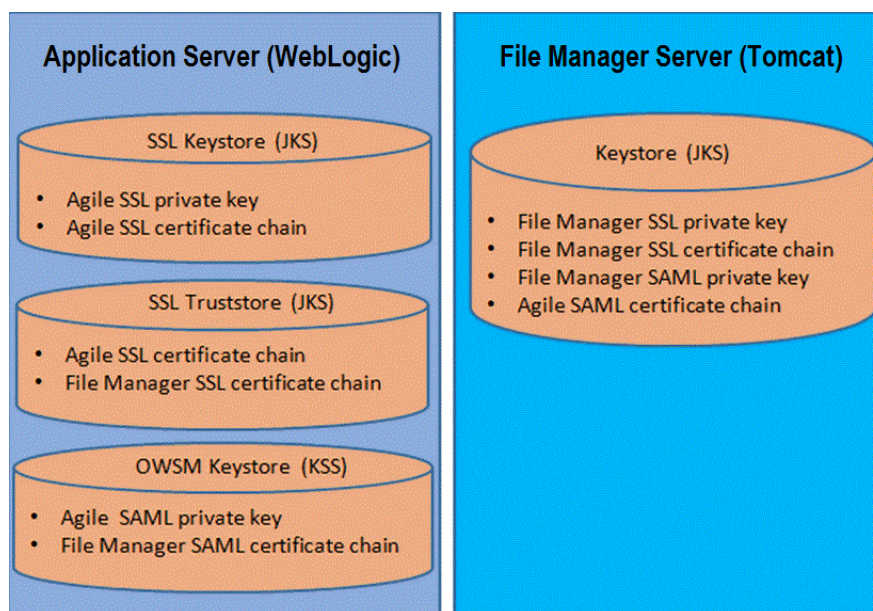
---

**Note:** SSL should be enabled as a prerequisite.

---

The following figure introduces the keystores/keys required for WSS configurations.

**Figure 7–1**



OWSM Keystore stores the Agile SAML key and File Manager SAML certificate. The File Manager keystore, which is same as the keystore storing the SSL key/certificates, stores the File Manager SAML key and Agile SAML certificate.

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 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\Agile936\Install\bin.
3. Execute the following command:

```
configureWSSecurity.cmd <RCU_DB_URL> <RCU_MDS_USER> <RCU_MDS_
USER_PASSWORD>
```

## 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.

---

**Note:** Starting with Agile PLM 9.3.6 RUP 4, you'll see a new menu with the options below:

- [1] Enable security policy
- [2] Enable client policy for Reference Objects
- [3] Enable client policy for CAD for Cloud

You'll see option [3] only if you're a user of CAD for Cloud.

Please choose [1] to enable the security policy for Agile PLM.

You can run WSSConfigurator tool again to enable the client policy for Reference Objects (and/or client policy for CAD for Cloud).

---

- 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:

```
936 WSS Service Configuration
file.server.saml.privatekey.alias=saml
file.server.saml.token.issuer=AgilePLM
```

---

**Note:** If you are using a CA certificate, the alias name must be the same as the alias in the CA certificate.

---

## Configuring Agile Server SAML Signature Key

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. It is a self-signed certificate, however, so we need to export it and request CA signature.

1. Launch the WSSConfigurator tool on the Agile Server and enter [4]: To manage OPSS OWSM keystore and press enter. For the second prompt, enter option [3]: Export Certificate and press enter. For the third prompt, enter option [2]: Certificate signing request.
2. Send the CSR to Certifying Authority to request a new certificate.
3. Receive the (A) root CA certificate, (B) Intermediate certificates, and (C) CA signed certificate.
4. Launch the WSSConfigurator tool on the Agile Server and enter [4]: To manage OPSS OWSM keystore and press enter. For the second prompt, enter option [2]: Import Trusted Certificate and press enter. For the third prompt, enter option [1]: Certificate. Import the signed certificates (C) CA signed certificate.

5. Enter [2]: Import Trusted Certificate and press enter. For the second prompt, enter option [2]: Trusted Certificate. Import the signed certificates (A) root CA certificate, (B) intermediate certificates.
6. Export the CA-signed saml certificate. Enter option [3] : Export Certificate, press enter. For the second prompt, enter option [1] : Certificate. Export as a9-saml-cert.cer.

## 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 and Certificate Signing Request

To generate the File Manager SAML Signature Key and Certificate Signing Request, do the following:

1. Generate SAML key in File Manager keystore.
  - Alias: saml
  - Key size: 2048
  - Algorithm: RSA
2. Generate the Certificate Signing Request with the SAML keystore above and send it to the Certifying Authority.
3. Import the CA certificate back to File Manager keystore.

### Import Agile Server SAML Signature Certificate into File Manager Keystore

You need to import the Agile Server SAML CA signed certificate, intermediate certificates and root CA certificate into the File Manager keystore.

1. Import the Agile Server SAML Signature certificate, a9-saml-cert.cer, into the File Manager SAML Keystore.
2. Import the intermediate certificate into File Manager Keystore SAML.
3. Import the root CA certificate into File Manager SAML Keystore.
4. Verify the keys/certificates in File Manager SAML Keystore.

### 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. Export the File Manager SAML certificate from fm-saml.jks, named as fm-ssl-cert.cer.
3. Execute the following command to change the certificate suffix from .cer to .der.  
`mv fm-saml-cert.cer fm-saml-cert.der`

4. 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`
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.
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'.

---

---

**Note:** If you are using a CA certificate, the alias name must be the same as the alias in the CA certificate.

---

Delete servers under <Agile\_Home>\agileDomain and then restart the WebLogic server and Tomcat server. Web Services Security is enabled for all A9 and File Manager 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	oracle/wss11_saml_or_username_token_with_message_protection_service_policy	oracle/wss_username_token_over_ssl_client_policy
AIS Service		
Reference Object		oracle/wss_saml_token_bearer_over_ssl_client_policy

## 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.jar"/>
```
  - b. Add the following Security Policy annotation in the Java file:
 

```
<referenceswsx>\src\com\agile\integration\externalreference\services\serv
ice\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\serv
ice\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.

---

## 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 File Manager startup script that were introduced for SSL configurations, as in section "[Configuring SSL on the File Manager](#)" on page 6-5.

### 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.



---

## 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.



## 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
Generate Agile Server SSL keystore and SSL key.		
Generate Certificate Signing Request to CA and import CA certificate back to Agile server SSL keystore.		
Create Agile Server truststore and import CA certificate chain.		
Enable SSL port in WebLogic console.		
Configure keystore and truststore in WebLogic console.		
Select SSL key as SSL Private key in Weblogic console.		
Change Java Client Web Server URL to HTTPS and change Port to SSL.		
Generate File Manager SSL keystore and SSL key.		
Generate Certificate Signing Request to CA and import CA certificate back to File Manager SSL keystore.		
Export Agile server SSL certificate and import it to File Manager SSL keystore.		
Export File Manager SSL certificate and import it to Agile SSL truststore.		
Change Server.xml to enable SSL.		

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

Task	Complete?	Comments
Change Server.conf to use HTTPs protocol and SSL ports.		
Extend WebLogic domain.		
Enable security using wssconfigurator.		
Generate Agile Server SAML signature key Certificate signing request using wssconfigurator and send it to CA.		
Import CA signed certificate back to OWSM keystore.		
Generate File Manager SAML key.		
Generate Certificate Signing Request to CA.		
Import Agile Server SAML CA signed certificate.		
Import CA certificate chain (in three layers) to File Manager SAML keystore.		
Import File Manager SAML Signature certificate to Agile Server keystore.		
Create Trust Issuer using wssconfigurator.		

## Distributed File Manager Configuration Checklist

**Table B–2 Checklist for DFM Configuration Tasks**

Task	Complete?	Comments
Generate DFM Keystore and SSL key.		
Generate Certificate Signing Request to CA and import CA certificate back to DFMs SSL keystore.		
Export DFMs SSL certificate and import to Agile Server truststore and FM SSL keystore.		
Change DFM server.conf with appropriate URLs.		
Import Agile SSL certificate and FM SSL certificate to DFMs SSL keystore.		
Import all FM SAML keys into DFM keystore (if SAML key is different in all FMs).		
Import DFMs SSL certificates into Agile server truststore and FM keystore.		
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) into Autovue/jre/security/cacerts.		



---

## 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*.

---

## 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.wls.jaxws-wls-wss-client.jar, which is under Weblogic\_HOME\wlserver\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 into Agile server keystore, similarly to Steps 5-7 mentioned in ["Import File Manager SAML Signature Certificate into Agile Server Keystore"](#) on page 7-4, but using the different alias name of "jsesignkey"  
  
If all is successful, there should be a certificate under owsm/keystore named 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.





---

# 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 8.5

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

```
<Connector protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="<certificate path>\fm-keystore.jks"
```

```
keystorePass=<keystore_password> keyAlias='<keystore_alias>'
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	<ol style="list-style-type: none"> <li>1. On the Internet Explorer Tools menu, click Internet Options.</li> <li>2. In the Internet Options dialog box, click the Advanced tab.</li> <li>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.</li> <li>4. Click OK.</li> <li>5. Exit and restart Internet Explorer.</li> </ol>
Mozilla Firefox	<ol style="list-style-type: none"> <li>1. Type about:config in the Firefox address bar and press Enter.</li> <li>2. Click I'll be careful, I promise!</li> <li>3. Type security.tls.version in the search bar.</li> <li>4. Double-click the preference of "security.tls.version.min" and set its value to 1.</li> <li>5. Restart Firefox.</li> </ol> <p>Alternatively, you can install the Firefox Extension SSL Version Control which provides a graphical way to specify the minimum SSL version.</p>
Chrome	<p>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 <code>Chrome.exe -ssl-version-min=tlsl</code> to specify that the minimum version of SSL to be used is TLS 1.0.</p>
Safari	<p>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.</p>



---

## Setting up AutoVue 21.0.1 for HTTPS

The section describes how to set up AutoVue 21.0.1 for HTTPS for various browsers.

### Internet Explorer/Google Chrome

Use Microsoft Windows built-in certificate store. If the certificate is imported for one browser, it will also be available for the other. Windows provides a certmgr utility, which can be used to remove certificates from the store.

1. Copy localhost.cer from the web accessible location to the client machine. For example, <AutoVue Install Dir>\tools\localhost.cer.
2. Double-click the localhost.cer file to open it.
3. Click the Install Certificate button. The Certificate Import wizard begins.
4. Click the Next button to run the import process.
5. Select the *Place all certificates in the following store* option and click the Browse button to select the Certificate Store.
6. Select the Trusted Root Certification Authorities store, click OK, then click the Next button.
7. When you click the Finish button to complete the process, a Windows popup dialog appears to warn about the risk to install such a certificate. Accept it with the Yes button:
8. Close the initial certificate dialog. The installation is complete.


### Mozilla Firefox

Mozilla Firefox uses its own certificate store instead of the Windows certificate store. The procedure to add a certificate for Firefox requires the use of additional tools. There are various tools that can be used to import a certificate into the Firefox certificate store. Refer to <https://wiki.mozilla.org/CA:AddRootToFirefox> for details.

As mentioned in the Mozilla wiki page, Mozilla is planning to accept the certificates stored in the Windows Store beginning from version 52, which is scheduled for March 7th, 2017. Then, the procedure described above for Chrome and Internet Explorer will work for Firefox, as well. Meanwhile, you can either use CCK2 Firefox Add-On or the certutil utility to import the certificate into the Firefox store. The following sections describe how to use each of these tools to import the certification in Firefox store.

### Using CCK2 to Import a Certificate

The following steps describe how to use the CCK2 tool to import a certificate:

1. Open Firefox and download a free version of CCK2.
2. Copy localhost.cer from the web location to the client.
3. Click the Allow button when prompted to allow the installation of the Add-On for your browser, then click Install, and then click the Restart Now button to restart Firefox.  
A new button  is added to the right side of the browser address bar.
4. Click on the new CCK2 button. A dialog window is displayed
5. Click the New button to create a new CCK2 configuration. A new window appears.
6. Fill in a Name and a unique Identifier.
7. In the next window, CCK2 wizard starts on the About page. Fill in the Description field, Version, and provide an Output Directory.
8. In the left pane, scroll down and select Certificates. The Certificates page appears in the wizard window.
9. Click the Add File button and browse for localhost.cer certificate. Select and add the file.
10. In the Certificates window, select all the three check-boxes with the Trust this CA to issue server certificates option. Click OK.
11. In the left pane of the main wizard window, scroll down to the last item in the list, Finish, and select it. The Finish page appears.
12. Click the Use AutoConfig button. A popup should confirm the successful creation of the CCK2 configuration with a message like "CCK2 Creation is complete and available at: <selected Output Directory>".
13. Close Firefox and the CCK2 Wizard. The following elements should be created under the selected output directory:
  - cck2.config.json file
  - autoconfig.zip
  - firefox folder
14. Unzip the contents of the autoconfig.zip file to the directory where Mozilla Firefox executable files are saved. This is typically C:\Program Files (x86)\Mozilla Firefox. The following files should be added to the Mozilla directory:
  - cck2.cfg file
  - cck2 folder
  - defaults folder
15. The certificate installation is complete. Restart Firefox and test.  
You can now remove the CCK2 Add-On and its output directory, if desired.

## Using Certutil to Import a Certificate

The following steps describe how to use the certutil utility tool to import a certificate:

1. Close the Firefox browser.

2. Download the Network Security Services Tool that contains the certutil utility from Mozilla. Unpack the ZIP file containing the certutil utility into a new directory on the client machine.
3. Copy the localhost.cer file from the web location to the new folder.
4. Find the cert8.db file under your Firefox profile,  
%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\xxxxxxx, in your home directory.
5. Open a command prompt window and go to the folder on which you unpacked the certutil utility. Run the following command:

```
certutil -A -d
%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\xxxxxxx -i
localhost.cer -n localhost -t "C,, "
```

---

**Note:** There is another certutil utility on Microsoft Windows which is in the path. Be careful to call the unpacked Mozilla certutil utility either directly from the scratch directory where it is or by appending explicitly its whole path to prevent calling the Windows certutil.

---



---

**Note:** To list Firefox certificates using this tool to confirm that you already installed it, run the following command:

---

```
certutil -L -d
%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\xxxxx
xxx
```

---



---

**Note:** To delete an existing certificate in order to replace it with the new one, run the following command:

---

```
certutil -D -d
%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\xx
xxxxxx -n localhost
```

---

6. Done. The certificate has been imported to Firefox successfully.

## Safari

To import a certificate into Safari browser, complete the following steps:

1. Start Safari.
2. Open the localhost.cer file in Safari from its web accessible location to download the certificate to the client. For example, <AutoVue Install Dir>\tools\localhost.cer.
3. Double-click the downloaded localhost.cer file. Keychain Access opens. Add the certificate to the keychain.
4. Double-click the added local host certificate. Ensure that the Trust menu is expanded and set the value of the Secure Sockets Layer (SSL) field to Always Trust.
5. Close Keychain Access. The certificate has been imported.

