CA SiteMinder® Web Services Security

WSS Agent for Oracle WebLogic Guide

12.52

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

CA Technologies Product References

This document references the following CA Technologies products:

- CA SiteMinder®
- CA SiteMinder® Web Services Security (formerly CA SOA Security Manager)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: SiteMinder WSS Agent for Oracle WebLogic Introduction 9

SiteMinder WSS Agent for Oracle WebLogic Overview ... 9
Required Background Information ... 11
SiteMinder WSS Agent for Oracle WebLogic Components ... 12
  SiteMinder WSS Agent JAX-RPC Handler ... 12
  SiteMinder WSS Agent Login Module ... 12
Installation Location References ... 13

Chapter 2: SiteMinder WSS Agent for WebLogic Install Preparation 15

Locate the Platform Support Matrix ... 15
Software Requirements ... 15
Installation Checklist ... 16
Preconfigure Policy Objects for SiteMinder WSS Agents ... 16
  Policy Object Preconfiguration Overview ... 17
  Preconfiguring the Policy Objects ... 18

Chapter 3: Install the SiteMinder WSS Agent for WebLogic on a Windows System 19

Set the JRE in the Path Variable ... 19
Apply the Unlimited Cryptography Patch to the JRE ... 19
Configure the JVM to Use the JSafeJCE Security Provider ... 20
Run the Installer to Install a SiteMinder WSS Agent ... 21
Install a SiteMinder WSS Agent Using the Unattended Installer ... 23
Installation and Configuration Log Files ... 24
How to Configure Agents and Register a System as a Trusted Host ... 24
  Gather Information Required for SiteMinder WSS Agent Configuration ... 25
  Configure a SiteMinder WSS Agent and Register a Trusted Host ... 26
Uninstall the SiteMinder WSS Agent ... 34

Chapter 4: Install the SiteMinder WSS Agent for WebLogic on a UNIX System 35

Set the JRE in the PATH Variable ... 35
Apply the Unlimited Cryptography Patch to the JRE ... 35
Configure the JVM to Use the JSafeJCE Security Provider ... 36
Run the Installer to Install a SiteMinder WSS Agent Using a GUI ... 37

Run the Installer to Install a SiteMinder WSS Agent Using a UNIX Console....39
Install a SiteMinder WSS Agent Using the Unattended Installer....41
Installation and Configuration Log Files....43
How to Configure Agents and Register a System as a Trusted Host....43
Gather Information Required for SiteMinder WSS Agent Configuration....44
Configure a SiteMinder WSS Agent and Register a Trusted Host....45
Uninstall the SiteMinder WSS Agent....52

Chapter 5: Upgrade a SOA Agent to a 12.52 WSS Agent 53

How to Upgrade a SOA Agent....53
Verify That the LD_PRELOAD Variable Does Not Conflict with Existing Agent....54
Run the Installation Wizard to Upgrade Your Agent on Windows....55
Run the Installation Wizard to Upgrade your Agent on UNIX/Linux....56
Set the Library Path Variable Before Configuring your Upgraded Agent on UNIX/Linux....57
Run the Configuration Wizard on Your Upgraded SiteMinder WSS Agent on Windows....57
Run the Configuration Wizard on Your Upgraded SiteMinder WSS Agent on UNIX/Linux....58

Chapter 6: Configure the SiteMinder WSS Agent 61

How to Configure the SiteMinder WSS Agent....61
SiteMinder WSS Agent for WebLogic Agent Configuration File....62
Agent Configuration Object....64
SiteMinder WSS Agent Configuration Parameters....64
Configure the Username and Password Digest Token Age Restriction....68

Chapter 7: Set the WebLogic Environment for the SiteMinder WSS Agent 69

WebLogic Environment Setting Locations....69
Set the WebLogic Environment on Windows....70
Set the WebLogic Environment on UNIX....71

Chapter 8: SiteMinder WSS Agent for Oracle WebLogic Logging 73

SiteMinder WSS Agent Logging....73
Log Files....73
SiteMinder WSS Agent Log....73
SiteMinder WSS Agent XML Message Processing Logging....74
Change the SiteMinder WSS Agent Log File Name....74
Append Messages to an Existing SiteMinder WSS Agent Log File....75
Set the SiteMinder WSS Agent File Log Level....75
Roll Over the SiteMinder WSS Agent Log File....75
Disable SiteMinder WSS Agent XML Message Processing Logging....76

SiteMinder WSS Agent Log Configuration File Summary 76
How to Set Log Files, and Command-line Help to Another Language 77
Determine the IANA Code for Your Language 78
Environment Variables 79

Chapter 9: Final Steps 83

Prevent WebLogic 10 from Loading Incompatible Version of XML Security 83
Restart WebLogic 83
Configure Web Services to Invoke the SiteMinder WSS Agent JAX-RPC Handler 83
Manually Edit JAX-RPC Web Service Deployment Descriptors 84
Use Handler Chains 85
Configure Policies for the SiteMinder WSS Agent 87
Troubleshoot Issues Related to Protected Web Services Making Use of Apache Commons Logging and Log4j 87
Commons Logging Class Loading Problem 88
log4j Conflict Problems 89

Chapter 1: SiteMinder WSS Agent for Oracle WebLogic Introduction

This section contains the following topics:

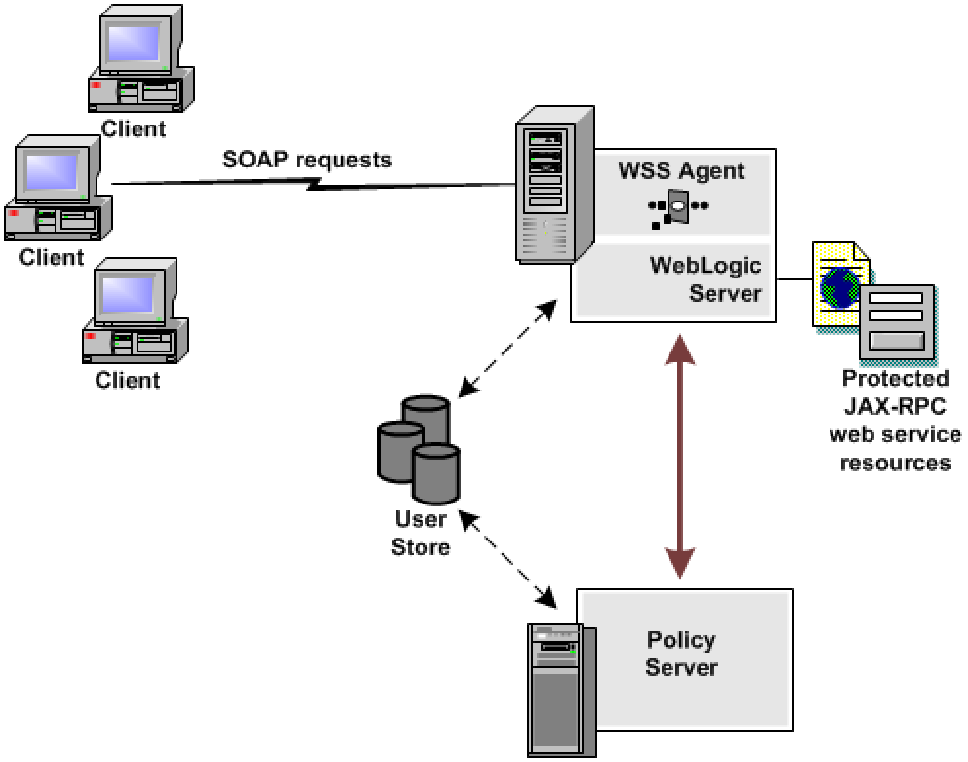
SiteMinder WSS Agent for Oracle WebLogic Overview (see page 9)
Required Background Information (see page 11)
SiteMinder WSS Agent for Oracle WebLogic Components (see page 12)
Installation Location References (see page 13)

SiteMinder WSS Agent for Oracle WebLogic Overview

The SiteMinder Web Services Security (WSS) Agent for Oracle WebLogic (formerly SOA Agent) resides in a WebLogic application server, enabling you to protect WebLogic-hosted JAX-RPC web service resources.

The SiteMinder WSS Agent for Oracle WebLogic intercepts all SOAP messages sent over HTTP(S) or JMS transports to JAX-RPC web services deployed on the WebLogic Server. The SiteMinder WSS Agent then communicates with the Policy Server to authenticate and authorize the message sender and, upon successful authentication and authorization, passes the SOAP message on to the addressed web service.

A high-level overview of the SiteMinder WSS Agent for Oracle WebLogic Server architecture is shown in the following figure.



The SiteMinder WSS Agent for Oracle WebLogic provides the following features:

- Fine-grained access control of JAX-RPC web service resources
- Support for bi-directional CA SiteMinder® Web Services Security/SiteMinder and WebLogic single sign-on (SSO)

The SiteMinder WSS Agent additionally supports:

- Multi-byte character user names
- Centralized and dynamic agent configurations
- Caching of resource protection decisions and authentication and authorization decisions
- Logging
- Authorization auditing

Required Background Information

This section is not intended for users who are new to Java, J2EE standards, or application server technology. It assumes that you have the following technical knowledge:

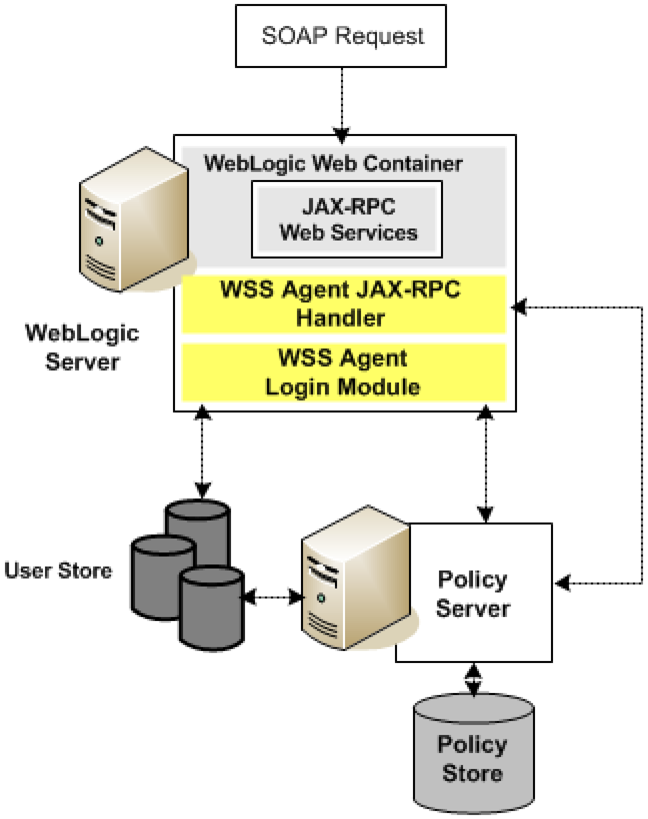
- An understanding of Java, J2EE standards, J2EE application servers, and multi-tier architecture
- An understanding of JAX-RPC web service implementations and JAX-RPC handlers
- Familiarity with the WebLogic Security Framework for WebLogic Server
- Familiarity with Java Authentication and Authorization Server (JAAS) and WebLogic security-related topics
- Experience with managing the WebLogic Server, including tasks such as accessing the administrative console
- Familiarity with CA SiteMinder® Web Services Security concepts, terms, and Policy Server configuration tasks

Additionally, to effectively plan your security infrastructure, you must be familiar with the web services that you plan to protect with CA SiteMinder® Web Services Security.

SiteMinder WSS Agent for Oracle WebLogic Components

The SiteMinder WSS Agent for Oracle WebLogic consists of two modules that plug into the WebLogic security infrastructure.

- SiteMinder WSS Agent JAX-RPC Handler
- SiteMinder WSS Agent Login Module



SiteMinder WSS Agent JAX-RPC Handler

The SiteMinder WSS Agent JAX-RPC Handler is a custom JAX-RPC Handler that, when added to the deployment descriptor of a JAX-RPC web service, intercepts SOAP message requests for JAX-RPC web services and diverts them to the SiteMinder WSS Agent Login Module for authentication and authorization decisions.

SiteMinder WSS Agent Login Module

The SiteMinder WSS Agent Login Module is a JAAS Login Module that performs authentication and authorization for JAX-RPC web services protected by the SiteMinder WSS Agent for Oracle WebLogic.

The SiteMinder WSS Agent Login Module authenticates credentials obtained from the following request types against associated user directories configured in CA SiteMinder® Web Services Security:

- SOAP requests intercepted by the SiteMinder WSS Agent JAX-RPC Handler .
- Requests for web service resources from users with pre-established CA SiteMinder® Web Services Security and SiteMinder sessions (validating the session and obtaining user names from associated SiteMinder session ticket cookies)

If CA SiteMinder® Web Services Security authentication is successful, the SiteMinder WSS Agent Login Module populates a JAAS Subject with a CA SiteMinder® Web Services Security Principal that contains the username and associated CA SiteMinder® Web Services Security session data. The SiteMinder WSS Agent Login Module then determines whether an authenticated user is allowed to access a protected WebLogic resource, based on associated CA SiteMinder® Web Services Security authorization policies.

Installation Location References

In this guide:

- *WSS_Home* refers to the location where CA SiteMinder® Web Services Security is installed.
- *WLS_HOME* refers to the installed location of the WebLogic Server.

Chapter 2: SiteMinder WSS Agent for WebLogic Install Preparation

This section contains the following topics:

Locate the Platform Support Matrix (see page 15)
Software Requirements (see page 15)
Installation Checklist (see page 16)
Preconfigure Policy Objects for SiteMinder WSS Agents (see page 16)

Locate the Platform Support Matrix

Use the Platform Support Matrix to verify that the operating environment and other required third-party components are supported.

Follow these steps:

1. Log in to the CA Support site.
2. Locate the Technical Support section.
3. Enter CA SiteMinder® in the Product Finder field.

   The CA SiteMinder® product page appears.
4. Click Product Status, CA SiteMinder® Family of Products Platform Support Matrices.

Note: You can download the latest JDK and JRE versions at the Oracle Developer Network.

Software Requirements

Before installing the SiteMinder WSS Agent for Oracle WebLogic, install the following software:

Note: Be sure to install the prerequisite software in the correct order.

- A supported version of Oracle WebLogic Server and any cumulative fixes for this application server. For WebLogic hardware and software requirements, see the WebLogic documentation.
- A supported Java Runtime Environment (JRE).
- A Policy Server

Note: The Policy Server can be installed on a different system than the WebLogic Server.

Note: For a list of supported CA and third-party components, refer to the CA SiteMinder® 12.52 Platform Support Matrix on the Technical Support site.

More information:

Locate the Platform Support Matrix (see page 15)

Installation Checklist

Before you install the SiteMinder WSS Agent for Oracle WebLogic on the WebLogic Server, complete the steps in the following table. To ensure proper configuration, follow the steps in order. You can place a check in the first column as you complete each step.

Completed?	Steps	For information, see...
	Install and configure the Policy Server.	*CA SiteMinder® Policy Server Installation Guide*
	Install the Oracle WebLogic Server.	The Oracle WebLogic Server Documentation
	Configure the Policy Server for the SiteMinder WSS Agent for Oracle WebLogic.	Preconfiguring Policy Objects for SiteMinder WSS Agents
	Install the SiteMinder WSS Agent on the Oracle WebLogic Server. Note: For WebLogic clusters, install the SiteMinder WSS Agent on each node in the cluster.	Install a SiteMinder WSS Agent on a Windows System or Install a SiteMinder WSS Agent on a UNIX System

Preconfigure Policy Objects for SiteMinder WSS Agents

This section describes how to preconfigure policy objects for SiteMinder WSS Agents on the Policy Server.

Policy Object Preconfiguration Overview

Before you install any SiteMinder WSS Agent, the CA SiteMinder® Web Services Security Policy Server must be installed and be able to communicate with the system where you plan to install the SiteMinder WSS Agent. Additionally, you must configure the Policy Server with the following:

- **An administrator that has the right to register trusted hosts**

  A trusted host is a client computer where one or more SiteMinder WSS Agents are installed. The term trusted host refers to the physical system. There must be an administrator with the privilege to register trusted hosts with the Policy Server.

  To configure an administrator, see the Administrators chapter of the *Policy Server Configuration Guide*.

- **Agent object/Agent identity**

  An Agent object creates an Agent identity by assigning the Agent a name. You define an Agent identity from the Agents object in the Administrative UI. You assign the Agent identity a name and specify the Agent type as a Web Agent.

  The name you assign for the Agent is the same name you specify in the DefaultAgentName parameter for the Agent Configuration Object that you must also define to centrally manage an Agent.

- **Host Configuration Object**

  This object defines the communication between the trusted host and the Policy Server after the initial connection between the two is made.

  A trusted host is a client computer where one or more SiteMinder WSS Agents can be installed. The term trusted host refers to the physical system, in this case the application server host.

  Do not confuse this object with the trusted host's configuration file, SmHost.conf, which is installed at the trusted host after a successful host registration. The settings in the SmHost.conf file enable the host to connect to a Policy Server for the first connection only. Subsequent connections are governed by the Host Configuration Object.

- **Agent Configuration Object**

  This object includes the parameters that define the SiteMinder Agent configuration. There are a few required parameters you must set for basic operation.

  The Agent Configuration Object must include a value for the DefaultAgentName parameter. This entry should match an entry you defined in the Agent object.

Note: For detailed information about how to configure SiteMinder WSS Agent-related objects, see the *Policy Server Configuration Guide*.

Preconfiguring the Policy Objects

The following is an overview of the configuration procedures you must perform on the Policy Server prior to installing the Agent software:

1. Duplicate or create a new Host Configuration Object, which holds initialization parameters for a Trusted Host. (If upgrading from an earlier Agent install, you can use the existing Host Configuration object).

   The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

2. As necessary, add or edit parameters in the Host Configuration Object that you just created.

3. Create an Agent identity for the SiteMinder WSS Agent. You must select **Web Agent** as the Agent type for the SiteMinder WSS Agent.

4. Duplicate an existing or create a new Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

5. In the Agent Configuration Object you just created, ensure that the DefaultAgentName parameter is set to specify the Agent identity defined in Step 3.

Chapter 3: Install the SiteMinder WSS Agent for WebLogic on a Windows System

This section contains the following topics:

Set the JRE in the Path Variable (see page 19)
Apply the Unlimited Cryptography Patch to the JRE (see page 19)
Configure the JVM to Use the JSafeJCE Security Provider (see page 20)
Run the Installer to Install a SiteMinder WSS Agent (see page 21)
Install a SiteMinder WSS Agent Using the Unattended Installer (see page 23)
Installation and Configuration Log Files (see page 24)
How to Configure Agents and Register a System as a Trusted Host (see page 24)
Uninstall the SiteMinder WSS Agent (see page 34)

Set the JRE in the Path Variable

Set the Java Runtime Environment (JRE) in the Windows path variable.

Follow these steps:

1. Open the Windows Control Panel.
2. Double-click System.
3. Add the location of the JRE to the Path system variable in the Environment Variables dialog.

Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows

  jre_home\lib\security

- UNIX

  jre_home/lib/security

jre_home

Defines the location of your Java Runtime Environment installation.

Note: If WebLogic has been configured to use its own JRE then you must also patch the JRE used by WebLogic to support unlimited key strength in the Java Cryptography Extension (JCE) package.

Configure the JVM to Use the JSafeJCE Security Provider

The SiteMinder WSS Agent XML encryption function requires that the JVM is configured to use the JSafeJCE security provider.

Follow these steps:

1. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE) to the java.security file located in the following location:

   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

   JVM_HOME

   Is the installed location of the JVM used by the application server.

   In the following example, the JSafeJCE security provider entry has been added as the second security provider:

   ```
   security.provider.1=sun.security.provider.Sun
   security.provider.2=com.rsa.jsafe.provider.JsafeJCE
   security.provider.3=sun.security.rsa.SunRsaSign
   security.provider.4=com.sun.net.ssl.internal.ssl.Provider
   security.provider.5=com.sun.crypto.provider.SunJCE
   security.provider.6=sun.security.jgss.SunProvider
   security.provider.7=com.sun.security.sasl.Provider
   ```

   Note: If using the IBM JRE, always configure the JSafeJCE security provider immediately after (that is with a security provider number one higher than) the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE)

2. Add the following line to *JVM_HOME*\jre\lib\security\java.security (Windows) or *JVM_HOME/*jre/lib/security/java.security (UNIX) to set the *initial* FIPS mode of the JsafeJCE security provider:

   ```
   com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
   ```

   Note: The initial FIPS mode does not affect the final FIPS mode you select for the SiteMinder WSS Agent.

Run the Installer to Install a SiteMinder WSS Agent

Install the SiteMinder WSS Agent using the CA SiteMinder® Web Services Security installation media on the Technical Support site.

Note: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.

To install a SiteMinder WSS Agent on Windows

1. Exit all applications that are running.
2. Navigate to the installation material.
3. Double-click ca-sm-wss-<SVMVER>-*cr*-win32.exe.

   cr

   Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

   The CA SiteMinder® Web Services Security installation wizard starts.

   Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA SiteMinder® Web Services Security Release Notes.

4. Use gathered system and component information to install the SiteMinder WSS Agent. Consider the following when running the installer:

   ■ When prompted to select what agents to install, select **CA SiteMinder® Web Services Security Agents for Application Servers** and then specify the **CA SiteMinder® Web Services Security Agent for Oracle WebLogic**.

   When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

   ■ When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):

   – WebLogic 9.2: C:\bea\weblogic92

   – WebLogic 10.3 C:\bea\wlserver_10.3

   – WebLogic 11G: C:\Oracle\Middleware\wlserver_10.3

   ■ If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   Note: If the installation program detects that newer versions of certain system DLLs are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The SiteMinder WSS Agent files are copied to the specified location.

6. On the CA SiteMinder® Web Services Security Configuration screen, click one of the following options and click Next:

   - Yes. I would like to configure CA SiteMinder® Web Services Security Agents now.
   - No. I will configure CA SiteMinder® Web Services Security Agents later.

   If the installation program detects that there are locked Agent files, it prompts you to restart your system instead of reconfiguring it. Select whether to restart the system automatically or later on your own.

7. Click Done.

   If you selected the option to configure SiteMinder WSS Agents now, the installation program prepares the CA SiteMinder® Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.

   If you installed a SiteMinder WSS Agent or Agents and did not select the option to configure SiteMinder WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

Installation Notes:

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log

  WSS_Home

  Specifies the path to where CA SiteMinder® Web Services Security is installed.

  Default: C:\Program Files\CA\Web Services Security

  install-date-and-time

  Specifies the date and time that the SiteMinder WSS Agent was installed.

  The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

More information:

How to Configure Agents and Register a System as a Trusted Host (see page 24)

Install a SiteMinder WSS Agent Using the Unattended Installer

After you have installed one or more SiteMinder WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall SiteMinder WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home*\install_config_info

WSS_Home

Specifies the path to where CA SiteMinder® Web Services Security is installed.

Default: C:\Program Files\CA\Web Services Security

To run the installer in the unattended installation mode

1. From a system where CA SiteMinder® Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.
2. Copy the SiteMinder WSS Agent installer file (ca-sm-wss-<SVMVER>-*cr*-win32.exe) into the same local directory as the ca-wss-installer.properties file.

   cr

   Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.
3. Open a console window and navigate to the location where you copied the files.
4. Run the following command:

   ```
   ca-sm-wss-<SVMVER>-cr-win32.exe -f ca-wss-installer.properties -i silent
   ```

   Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA SiteMinder® Web Services Security Release Notes.

   The -i silent setting instructs the installer to run in the unattended installation mode.

   Note: If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   Example:

   ```
   ca-sm-wss-<SVMVER>-cr-win32.exe -f "C:\Program Files\CA\Web Services
   Security\install_config_info\ca-wss-installer.properties" -i silent
   ```

An InstallAnywhere status bar appears, which shows that the unattended CA SiteMinder® Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

Installation Notes:

- After installation, you can review the installation log file in *WSS_HOME*\install_config_info. The file name is: CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log

  WSS_Home

  Specifies the path to where CA SiteMinder® Web Services Security is installed.

  Default: C:\Program Files\CA\Web Services Security

  install-date-and-time

  Specifies the date and time that the SiteMinder WSS Agent was installed.

  The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

- To stop the installation manually, type Ctrl+C.

Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a SiteMinder WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time*_InstallLog.log file located in *WSS_Home*\install_config_info.

date-time

Specifies the date and time of the CA SiteMinder® Web Services Security installation.

How to Configure Agents and Register a System as a Trusted Host

A *trusted host* is a client computer where one or more SiteMinder WSS Agents can be installed. The term trusted host refers to the physical system.

To establish a connection between the trusted host and the Policy Server, register the host with the Policy Server. When registration is complete the SmHost.conf file is created. After this file is created successfully, the client computer becomes a trusted host.

Gather Information Required for SiteMinder WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

SM Admin User Name

The name of a Policy Server administrator allowed to register the host with the Policy Server.

This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator is SiteMinder.

SM Admin Password

The Policy Server administrator account password.

Trusted Host Name

Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

Note: This name must be unique among trusted hosts and not match the name of any other Agent.

Host Configuration Object

The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

Note: This value must match the Host Configuration Object entry preconfigured on the Policy Server.

Policy Server IP Address

The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used.

You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:

Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)

Note also that if you specify a non-default port, that port is used for the Policy Server’s authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:

policyserver="*ip_address*,5555,5555,5555"

FIPS Encryption Mode

Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

FIPS Compatibility Mode (Default)

Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA SiteMinder® encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

FIPS Only Mode

Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.

Important! A CA SiteMinder® installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA SiteMinder®, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

Configure a SiteMinder WSS Agent and Register a Trusted Host

You configure a SiteMinder WSS Agent and register the system that hosts it as a trusted host using the CA SiteMinder® Web Services Security Configuration Wizard.

Configure an Agent and Register Your System as a Trusted Host on Windows

You can configure your SiteMinder WSS Agent and register a trusted host immediately after installing the SiteMinder WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

Note: You only register the host once, *not* each time you install and configure a SiteMinder WSS Agent on your system.

Follow these steps:

1. Open the following directory on your web server:

   ```
   WSS_Home\install_config_info
   ```

   WSS_Home

   Specifies the path to where CA SiteMinder® Web Services Security is installed.

   Default: C:\Program Files\CA\Web Services Security

2. Right-click ca-pep-config.exe, and then select Run as administrator.

   Important! If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA SiteMinder® component.

   The WSS Agent Configuration Wizard starts.

3. Use gathered system and component information to configure the SiteMinder WSS Agent and register the host.

   Note: If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*\config. You can modify this file.

agent_home

Is the installed location of the SiteMinder WSS Agent.

Modify the SmHost.conf File (Windows)

SiteMinder WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

To modify the SmHost.conf file

1. Navigate to the *agent_home*\config directory.
2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

   Important! Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

   hostconfigobject

   > Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.
   >
   > If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

   Example: **hostconfigobject="*host_configuration_object*"**

   policyserver

   > Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:
   >
   > "*IP_address*, *port*,*port*,*port*"
   >
   > The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.
   >
   > To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.
   >
   > Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA SiteMinder® environment or is no longer in service, delete the entry.
   >
   > **Important:** If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.
   >
   > **Default:** *IP_address*, 44441,44442,44443
   >
   > **Example** (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"
   >
   > **Example** (Syntax for multiple entries, place each Policy Server on a separate line)**:**
   > policyserver="123.122.1.1, 44441,44442,44443"
   > policyserver="111.222.2.2, 44441,44442,44443"
   > policyserver="321.123.1.1, 44441,44442,44443"

requesttimeout

Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

Default: 60

Example: requesttimeout="60"

4. Save and close the SmHost.Conf file.

   The changes to the SmHost.conf file are applied.

Re-register a Trusted Host Using the Registration Tool (Windows)

When you install a SiteMinder WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA SiteMinder® environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home*\bin directory when you install a SiteMinder WSS Agent.

agent_home

Is the installed location of the SiteMinder WSS Agent.

To re-register a trusted host using the registration tool

1. Open a command prompt window.
2. Enter the smreghost command using the following required arguments:

```
smreghost -i policy_server_IP_address:[port]
-u administrator_username -p Administrator_password
-hn hostname_for_registration -hc host_configuration_ object
```

Note: Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

See the following example:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings
```

The following example contains the -o argument:

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
-hc DefaultHostSettings -o
```

The following arguments are used with the smreghost command:

-i *policy_server_IP_ address:port*

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

Default: (ports) 44441,44442,44443

Example: (IPv4 non-default port of 55555) -i 127.0.0.1:55555

Example: (IPv4 default ports) -i 127.0.0.1

Example: (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

Example: (IPv6 default ports) -i [2001:DB8::/32]

-u *administrator_username*

Indicates the name of the CA SiteMinder® administrator with the rights to register a trusted host.

-p *Administrator_password*

Indicates the password of the Administrator who is allowed to register a trusted host.

-hn *hostname_for_registration*

Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

-hc *host_config_object*

Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server before you can register a trusted host.

-sh *shared_secret*

Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

-rs

Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

-f *path_to_host_config_file*

(Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

-cf *FIPS mode*

Specifies one of the following FIPS modes:

- COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA SiteMinder® encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.
- ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

Important! A CA SiteMinder® installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA SiteMinder®, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

Default: COMPAT

Note: More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA SiteMinder® Cryptographic Boundary exists in the *Policy Server Administration Guide*.

Note: More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the SiteMinder Cryptographic Boundary exists in the Policy Server Administration Guide.

-o

Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

Register Multiple Trusted Hosts on One System (Windows)

You typically register only one trusted host for each machine where web servers and Agents are installed. However, you can register multiple trusted hosts on one computer to create distinct connections for each CA SiteMinder® client. Using multiple trusted hosts ensures a unique shared secret and a secure connection for each client requiring communication with the Policy Server.

For most installations this is not a recommended configuration. However, it is an option for sites who require distinct, secure channels for each client or group of client applications protected by CA SiteMinder® Agents. For example, an application service provider may have many client computers with different applications installed. You may want a secure connection for each application, which you can achieve by registering multiple trusted hosts. The Policy Server then issues unique shared secrets for each client connection.

To register multiple trusted hosts, use one of the following methods:

- Registering with the Configuration Wizard: To register additional servers as trusted hosts, go through the registration process again; however, when prompted to specify a location for the SmHost.conf file, enter a unique path. Do not register a new host and use an existing web server's SmHost.conf file or that file will be overwritten. You can use the name SmHost.conf or give the file a new name.

  Important! If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA SiteMinder® component.

  Note: If you have registered a trusted host with a Policy Server and you run the Configuration Wizard to configure subsequent Agents without using a unique path for the SmHost.conf file, you will see a warning message in the Host Registration dialog box. The message reads:

  "Warning: You have already registered this Agent with a Policy Server."

- Registering with the smreghost command-line tool: Run the smreghost tool after you have completed the first Agent installation on a given computer. You can run this tool for each trusted host that you want to register.

  Important! Before running a CA SiteMinder® utility or executable on Windows Server 2008, open the command line window with administrator permissions. Open the command line window this way, even if your account has administrator privileges.

Uninstall the SiteMinder WSS Agent

To uninstall the SiteMinder WSS Agent, run the CA SiteMinder® Web Services Security uninstall wizard.

Follow these steps:

1. Navigate to the *WSS_HOME*\install_config_info (Windows) or *WSS_HOME*/install_config_info (UNIX) directory and run the CA SiteMinder® Web Services Security uninstall wizard to remove CA SiteMinder® Web Services Security agents:

   ■ Windows: soa-uninstall.cmd

   ■ UNIX: soa-uninstall.sh

   WSS_HOME

   Specifies the CA SiteMinder® Web Services Security installation location.

   Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA SiteMinder® Web Services Security Release Notes.

   The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.

3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.

   The uninstall wizard removes all selected CA SiteMinder® Web Services Security components.

4. Restart the server.

Chapter 4: Install the SiteMinder WSS Agent for WebLogic on a UNIX System

This section contains the following topics:

Set the JRE in the PATH Variable (see page 35)
Apply the Unlimited Cryptography Patch to the JRE (see page 35)
Configure the JVM to Use the JSafeJCE Security Provider (see page 36)
Run the Installer to Install a SiteMinder WSS Agent Using a GUI (see page 37)
Run the Installer to Install a SiteMinder WSS Agent Using a UNIX Console (see page 39)
Install a SiteMinder WSS Agent Using the Unattended Installer (see page 41)
Installation and Configuration Log Files (see page 43)
How to Configure Agents and Register a System as a Trusted Host (see page 43)
Uninstall the SiteMinder WSS Agent (see page 52)

Set the JRE in the PATH Variable

Set the Java Runtime Environment (JRE) in the UNIX system PATH variable.

To set the JRE in the PATH variable

1. Open a Command Window.
2. Run the following commands:

```
PATH=$PATH:JRE
export PATH
```

JRE

Defines the location of your Java Runtime Environment bin directory.

Apply the Unlimited Cryptography Patch to the JRE

Patch the Java Runtime Environment (JRE) used by the Agent to support unlimited key strength in the Java Cryptography Extension (JCE) package. The patches for all supported platforms are available from the Oracle website.

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

The local_policy.jar and US_export_policy.jar files can found be in the following locations:

- Windows

  jre_home\lib\security

- UNIX

  jre_home/lib/security

jre_home

Defines the location of your Java Runtime Environment installation.

Note: If WebLogic has been configured to use its own JRE then you must also patch the JRE used by WebLogic to support unlimited key strength in the Java Cryptography Extension (JCE) package.

Configure the JVM to Use the JSafeJCE Security Provider

The SiteMinder WSS Agent XML encryption function requires that the JVM is configured to use the JSafeJCE security provider.

Follow these steps:

1. Add a security provider entry for JSafeJCE (com.rsa.jsafe.provider.JsafeJCE) to the java.security file located in the following location:

   - *JVM_HOME*\jre\lib\security (Windows)
   - *JVM_HOME*/jre/lib/security (UNIX)

   JVM_HOME

   Is the installed location of the JVM used by the application server.

In the following example, the JSafeJCE security provider entry has been added as the second security provider:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.rsa.jsafe.provider.JsafeJCE
security.provider.3=sun.security.rsa.SunRsaSign
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
security.provider.7=com.sun.security.sasl.Provider
```

Note: If using the IBM JRE, always configure the JSafeJCE security provider immediately after (that is with a security provider number one higher than) the IBMJCE security provider (com.ibm.crypto.provider.IBMJCE)

2. Add the following line to *JVM_HOME*\jre\lib\security\java.security (Windows) or *JVM_HOME/*jre/lib/security/java.security (UNIX) to set the *initial* FIPS mode of the JsafeJCE security provider:

```
com.rsa.cryptoj.fips140initialmode=NON_FIPS140_MODE
```

Note: The initial FIPS mode does not affect the final FIPS mode you select for the SiteMinder WSS Agent.

Run the Installer to Install a SiteMinder WSS Agent Using a GUI

Install the SiteMinder WSS Agent using the CA SiteMinder® Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-<SVMVER>-cr-unix_version.bin
  ```

  cr

  Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

  unix_version

  Specifies the UNIX version: **sol** or **linux**.

- If you execute the CA SiteMinder® Web Services Security installer across different subnets, it can crash. Install CA SiteMinder® Web Services Security components directly on the host system to avoid the problem.

To install the SiteMinder WSS Agent

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.

3. Enter the following command:

   ```
   ./ca-sm-wss-<SVMVER>-cr-unix_version.bin
   ```

   The CA SiteMinder® Web Services Security installer starts.

4. Use gathered system and component information to install the SiteMinder WSS Agent. Consider the following when running the installer:

   - When prompted to select what agents to install, select **CA SiteMinder® Web Services Security Agents for Application Servers** and then specify the **CA SiteMinder® Web Services Security Agent for Oracle WebLogic**.

   – When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).

   - When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):

     - WebLogic 9.2: /bea/weblogic92
     - WebLogic 10.3 /bea/wlserver_10.3
     - WebLogic 11G: /Oracle/Middleware/wlserver_10.3

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.

   - Do not use space characters in the SiteMinder WSS Agent install path. For example, "/CA Technologies/agent" will result in install failure.

5. Review the information presented on the Pre-Installation Summary page, then click Install.

   Note: If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The SiteMinder WSS Agent files are copied to the specified location. Afterward, the CA CA SiteMinder® Web Services Security Configuration screen is displayed.

6. Select one of the following options:

   - Yes. I would like to configure CA SiteMinder® Web Services Security Agents now.

   - No. I will configure CA SiteMinder® Web Services Security Agents later.

7. Click Done.

   If you selected the option to configure SiteMinder WSS Agents now, the installation program prepares the CA SiteMinder® Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.

   If you did not select the option to configure SiteMinder WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

Installation Notes:

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/*install_config_info directory. This log file contains the results of the installation.

  WSS_Home

  Specifies the path to where CA SiteMinder® Web Services Security is installed.

  install-date-and-time

  Specifies the date and time that the SiteMinder WSS Agent was installed.

- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

More information:

How to Configure Agents and Register a System as a Trusted Host (see page 24)

Run the Installer to Install a SiteMinder WSS Agent Using a UNIX Console

Install the SiteMinder WSS Agent using the CA SiteMinder® Web Services Security installation media on the Technical Support site. Consider the following:

- Depending on your permissions, you may need to add executable permissions to the install file by running the following command:

  ```
  chmod +x ca-sm-wss-<SVMVER>-cr-unix_version.bin
  ```

  cr

  Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

  unix_version

  Specifies the UNIX version: **sol** or **linux**..

- If you execute the CA SiteMinder® Web Services Security installer across different subnets, it can crash. Install CA SiteMinder® Web Services Security components directly on the host system to avoid the problem.

To install the SiteMinder WSS Agent

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located.
3. Enter the following command:

   ```
   ./ca-sm-wss-<SVMVER>-cr-unix_version.bin -i console
   ```

   The CA SiteMinder® Web Services Security installer starts.
4. Use gathered system and component information to install the SiteMinder WSS Agent. Consider the following as you make your selections:
   - When prompted to select what agents to install, select **CA SiteMinder® Web Services Security Agents for Application Servers** and then specify the **CA SiteMinder® Web Services Security Agent for Oracle WebLogic**.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - When prompted for the location where WebLogic is installed, enter one of the following locations (as appropriate for your version of WebLogic):
     - WebLogic 9.2: /bea/weblogic92
     - WebLogic 10.3   /bea/wlserver_10.3
     - WebLogic 11G: /Oracle/Middleware/wlserver_10.3
   - When prompted for the location where WebLogic is installed, enter the correct location for your version of WebLogic.
   - Do not use space characters in the SiteMinder WSS Agent install path. For example, "/CA Technologies/agent" will result in install failure.
5. Review the information presented on the Pre-Installation Summary page, then proceed.

   Note: If the installation program detects that newer versions of certain system libraries are installed on your system it asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The SiteMinder WSS Agent files are copied to the specified location. Afterward, the CA CA SiteMinder® Web Services Security Configuration screen is displayed.

6. Select one of the following options:
   - Yes. I would like to configure CA SiteMinder® Web Services Security Agents now.
   - No. I will configure CA SiteMinder® Web Services Security Agents later.
7. Hit Enter.

   If you selected the option to configure SiteMinder WSS Agents now, the installation program prepares the CA SiteMinder® Web Services Security Configuration Wizard and begins the trusted host registration and configuration process.

   If you did not select the option to configure SiteMinder WSS Agents now or if you are required to reboot the system after installation you must start the configuration wizard manually later.

Installation Notes:

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/*install_config_info directory. This log file contains the results of the installation.

  WSS_Home

  Specifies the path to where CA SiteMinder® Web Services Security is installed.

  install-date-and-time

  Specifies the date and time that the SiteMinder WSS Agent was installed.

  The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.

More information:

How to Configure Agents and Register a System as a Trusted Host (see page 24)

Install a SiteMinder WSS Agent Using the Unattended Installer

After you have installed one or more SiteMinder WSS Agents on one machine, you can reinstall those agents on the same machine or install them with the same options on another machine using an unattended installation mode. An unattended installation lets you install or uninstall SiteMinder WSS Agents without any user interaction

The unattended installation uses the ca-wss-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, paths, and so on.

The ca-wss-installer.properties file is located in: *WSS_Home/*install_config_info

WSS_Home

Specifies the path to where CA SiteMinder® Web Services Security is installed.

Default: C:\Program Files\CA\Web Services Security

To run the installer in the unattended installation mode

1. From a system where CA SiteMinder® Web Services Security is already installed, copy the ca-wss-installer.properties file to a local directory on your system.
2. Copy the SiteMinder WSS Agent installer file (ca-sm-wss-<SVMVER>-*cr*-*unix_version*) into the same local directory as the ca-wss-installer.properties file.

   cr

   Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

   unix_version

   Specifies the UNIX version: **sol** or **linux**.
3. Open a console window and navigate to the location where you copied the files.
4. Run the following command:

   ```
   ./ca-sm-wss-<SVMVER>-cr-unix_version -f ca-wss-installer.properties -i silent
   ```

   The -i silent setting instructs the installer to run in the unattended installation mode.

   Note: If the ca-wss-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   Example:

   ```
   ./ca-sm-wss-<SVMVER>-cr-unix_version -f
   ~/CA/Web_Services_Security/install_config_info/ca-wss-installer.properties"
   -i silent
   ```

   An InstallAnywhere status bar appears, which shows that the unattended CA SiteMinder® Web Services Security installer has begun. The installer uses the parameters specified in the ca-wss-installer.properties file.

Installation Notes:

- To check if the unattended installation completed successfully, see the CA_SiteMinder_Web_Services_Security_Install_*install-date-and-time*.log file in *WSS_HOME/*install_config_info directory. This log file contains the results of the installation.

  WSS_Home

  Specifies the path to where CA SiteMinder® Web Services Security is installed.

  install-date-and-time

  Specifies the date and time that the SiteMinder WSS Agent was installed.

- The Agent cannot communicate properly with the Policy Server until the trusted host is registered.
- To stop the installation manually, type Ctrl+C.

Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a SiteMinder WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time*_InstallLog.log file located in *WSS_Home*\install_config_info.

date-time

Specifies the date and time of the CA SiteMinder® Web Services Security installation.

How to Configure Agents and Register a System as a Trusted Host

A *trusted host* is a client computer where one or more SiteMinder WSS Agents can be installed. The term trusted host refers to the physical system.

To establish a connection between the trusted host and the Policy Server, register the host with the Policy Server. When registration is complete the SmHost.conf file is created. After this file is created successfully, the client computer becomes a trusted host.

Gather Information Required for SiteMinder WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

SM Admin User Name

The name of a Policy Server administrator allowed to register the host with the Policy Server.

This administrator should already be defined at the Policy Server and have the permission Register Trusted Hosts set. The default administrator is SiteMinder.

SM Admin Password

The Policy Server administrator account password.

Trusted Host Name

Specifies a unique name that represents the trusted host to the Policy Server. This name *does not* have to be the same as the physical client system that you are registering; it can be any unique name, for example, mytrustedhost.

Note: This name must be unique among trusted hosts and not match the name of any other Agent.

Host Configuration Object

The name of the Host Configuration Object in the Policy Server that defines the connection between the trusted host and the Policy Server. For example, to use the default, enter DefaultHostSettings. In most cases, you will have created your own Host Configuration Object.

Note: This value must match the Host Configuration Object entry preconfigured on the Policy Server.

Policy Server IP Address

The IP address, or host name, and authentication port of the Policy Server where you are registering the host. The default port is 44442. If you do not provide a port, the default is used.

You can specify a non-default port number, but if your Policy Server is configured to use a non-default port and you omit it when you register a trusted host, the following error is displayed:

Registration Failed (bad ipAddress[:port] or unable to connect to Authentication server (-1)

Note also that if you specify a non-default port, that port is used for the Policy Server’s authentication, authorization, and accounting ports; however, the unified server responds to any Agent request on any port. The entry in the SmHost.conf file will look like:

policyserver="*ip_address*,5555,5555,5555"

FIPS Encryption Mode

Determines whether the Agent communicates with the Policy Server using certified Federal Information Processing Standard (FIPS) 140-2 compliant cryptographic libraries.

FIPS Compatibility Mode (Default)

Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA SiteMinder® encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.

FIPS Only Mode

Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using only FIPS 140-2 algorithms.

Important! A CA SiteMinder® installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA SiteMinder®, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

Configure a SiteMinder WSS Agent and Register a Trusted Host

You configure a SiteMinder WSS Agent and register the system that hosts it as a trusted host using the CA SiteMinder® Web Services Security Configuration Wizard.

Run the SiteMinder WSS Agent Configuration Program on UNIX or Linux Systems

You can configure your SiteMinder WSS Agents and register a trusted host immediately after installing the SiteMinder WSS Agent or at a later time; however, the host must be registered to communicate with the Policy Server.

Note: You only register the host once, *not* each time you install and configure a SiteMinder WSS Agent on your system.

These instructions are for GUI and Console Mode registration. The steps for the two modes are the same, with the following exceptions for Console mode:

- You may be instructed to select an option by entering a corresponding number for that option.

- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

To configure Agents and register a trusted host

1. If necessary, start the Configuration Wizard as follows:

   a. Open a console window.

   b. Navigate to *agent_home*/install_config_info, where *agent_home* is the installed location of the SiteMinder WSS Agent.

   c. Enter one of the following commands:

      GUI Mode: ./ca-pep-config.bin

      Console Mode: ./ca-pep-config.bin -i console

   The Configuration Wizard starts.

2. Use gathered system and component information to configure the SiteMinder WSS Agent and register the host.

   Note: If you choose to configure multiple Agents, you can set the Register with same Policy Server option to register them all with the same Policy Server.

When the wizard completes, the host is registered and a host configuration file, SmHost.conf, is created in *agent_home*/config. You can modify this file.

agent_home

Is the installed location of the SiteMinder WSS Agent.

Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of a SiteMinder WSS Agent, check the CA_SiteMinder_Web_Services_Security_Install_*date-time*_InstallLog.log file located in *WSS_Home*\install_config_info.

date-time

Specifies the date and time of the CA SiteMinder® Web Services Security installation.

Modify the SmHost.conf File

SiteMinder WSS Agents act as trusted hosts by using the information in the SmHost.conf file to locate and make initial connections to a Policy Server. Once the Agent connects to the Policy Server, the initial connections are closed. Any further communication between the Agent and the Policy Server is based on settings in the Host Configuration Object that is located on the Policy Server.

You can modify portions of the SmHost.conf file to change the initial Agent-to-Policy Server connection.

To modify the SmHost.conf file

1. Navigate to the *agent_home*/config directory.

   agent_home

   Is the installed location of the SiteMinder WSS Agent.

2. Open the SmHost.conf file in a text editor.

3. Enter new values for the any of the following settings that you want to change:

   Important! Change only the settings of the parameters listed here. Do not modify the settings of any other parameters in the SmHost.conf file.

   hostconfigobject

   Specifies the host configuration object that defines connectivity between the Agent that is acting as trusted host and the Policy Server. This name must match a name defined in the Administrative UI.

   If you want to change the host configuration object an object so the SOA Agent uses it, you need to modify this setting.

   Example: **hostconfigobject="*host_configuration_object*"**

   policyserver

   Specifies the Policy Server to which the trusted host will try to connect. The proper syntax is as follows:

   "*IP_address*, *port*,*port*,*port*"

   The default ports are 44441,44442,44443, but you can specify non-default ports using the same number or different numbers for all three ports. The unified server responds to any Agent request on any port.

   To specify additional bootstrap servers for the Agent, add multiple Policy Server entries to the file. Multiple entries provide the Agent with several Policy Servers to which it can connect to retrieve its Host Configuration Object. After the Host Configuration Object is retrieved, the bootstrap servers are no longer needed for that server process.

   Multiple entries can be added during host registration or by modifying this parameter. If a Policy Server is removed from your CA SiteMinder® environment or is no longer in service, delete the entry.

   Important: If an Agent is configured on a multi-process web server, specifying multiple Policy Server entries is recommended to ensure that any child process can establish a connection to the secondary Policy Server if the primary Policy Server fails. Each time a new child process is started, it will not be able to initialize the Agent if only one Policy Server is listed in the file and that Policy Server is unreachable.

Default: *IP_address*, 44441,44442,44443

Example (Syntax for a single entry)**:** "*IP_address*, *port*,*port*,*port*"

Example (Syntax for multiple entries, place each Policy Server on a separate line)**:**
policyserver="123.122.1.1, 44441,44442,44443"
policyserver="111.222.2.2, 44441,44442,44443"
policyserver="321.123.1.1, 44441,44442,44443"

requesttimeout

Specifies an interval of seconds during which the Agent that is acting as a trusted host waits before deciding that a Policy Server is unavailable. You can increase the time-out value if the Policy Server is busy due to heavy traffic or a slow network connection.

Default: 60

Example: requesttimeout="60"

4. Save and close the SmHost.Conf file.

   The changes to the SmHost.conf file are applied.

Re-register a Trusted Host Using the Registration Tool (UNIX)

When you install a SiteMinder WSS Agent on a server for the first time, you are prompted to register that server as a trusted host. After the trusted host is registered, you do not have to re-register with subsequent agent installations. There are some situations where you may need to re-register a trusted host independently of installing an Agent, such as the following:

- To rename the trusted host if there has been a change to your CA SiteMinder® environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.
- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

The registration tool, smreghost, re-registers a trusted host. This tool is installed in the *agent_home*/bin directory when you install a SiteMinder WSS Agent.

agent_home

Is the installed location of the SiteMinder WSS Agent.

To re-register a trusted host using the registration tool

1. Open a command prompt window.
2. Ensure that the library path environment variable contains the path to the agent bin directory.
3. Enter the following two commands:

   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:*agent_home*/bin

   export LD_LIBRARY_PATH

   For example, for a SiteMinder WSS Agent for WebLogic, enter the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/SOA Security Manager/wlsagent/bin
   export LD_LIBRARY_PATH
   ```

4. Enter the smreghost command using the following required arguments:

   ```
   smreghost -i policy_server_IP_address:[port]
   -u administrator_username -p Administrator_password
   -hn hostname_for_registration -hc host_configuration_ object
   ```

   Note: Separate each command argument from its value with a space. Surround any values that contain spaces with double quotes (").

   See the following example:

   ```
   smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
   -hc DefaultHostSettings
   ```

   The following example contains the -o argument:

   ```
   smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn "host computer A"
   -hc DefaultHostSettings -o
   ```

The following arguments are used with the smreghost command:

-i ***policy_server_IP_ address:port***

Indicates the IP address of the Policy Server where you are registering this host. Specify the port of the authentication server only if you are *not* using the default port.

If you specify a port number, which can be a non-default port, that port is used for all three Policy Server processes (authentication, authorization, accounting). The Policy Server responds to any Agent request on any port.

Use a colon between the IP address and non-default port number, as shown in the following examples.

Default: (ports) 44441,44442,44443

Example: (IPv4 non-default port of 55555) -i 127.0.0.1:55555

Example: (IPv4 default ports) -i 127.0.0.1

Example: (IPv6 non-default port of 55555) -i [2001:DB8::/32][:55555]

Example: (IPv6 default ports) -i [2001:DB8::/32]

-u ***administrator_username***

Indicates the name of the CA SiteMinder® administrator with the rights to register a trusted host.

-p ***Administrator_password***

Indicates the password of the Administrator who is allowed to register a trusted host.

-hn ***hostname_for_registration***

Indicates the name of the host to be registered. This can be any name that identifies the host, but it must be unique. After registration, this name is placed in the Trusted Host list in the Administrative UI.

-hc ***host_config_object***

Indicates the name of the Host Configuration Object configured at the Policy Server. This object must exist on the Policy Server before you can register a trusted host.

-sh ***shared_secret***

Specifies the shared secret for the agent, which is stored in the SmHost.conf file on the local web server. This argument changes the shared secret on only the local web server. The Policy Server is not contacted.

-rs

Specifies whether the shared secret will be updated (rolled over) automatically by the Policy server. This argument instructs the Policy Server to update the shared secret.

-f *path_to_host_config_file*

(Optional) Indicates the full path to the file that contains the registration data. The default file is SmHost.conf. If you do not specify a path, the file is installed in the location where you are running the smreghost tool.

If you use the same name as an existing host configuration file, the tool backs up the original and adds a .bk extension to the backup file name.

-cf *FIPS mode*

Specifies one of the following FIPS modes:

- COMPAT--Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing CA SiteMinder® encryption algorithms. If your organization does not require the use of FIPS-compliant algorithms, the Policy Server and the Agents can operate in non-FIPS mode without further configuration.
- ONLY--Specifies full-FIPS mode, which requires that the Policy Server and Web Agents read and write information using *only* FIPS 140-2 algorithms.

Important! A CA SiteMinder® installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of CA SiteMinder®, including all agents, custom software using older versions of the Agent API, and custom software using PM APIs or any other API that the Policy Server exposes. You must re-link all such software with the corresponding versions of the respective SDKs to achieve the required support for Full FIPS mode.

If this switch is not used, or you use the switch without specifying a mode, the default setting is used.

Default: COMPAT

Note: More information on the FIPS Certified Module and the algorithms being used; the data that is being protected; and the CA SiteMinder® Cryptographic Boundary exists in the *Policy Server Administration Guide*.

-o

Overwrites an existing trusted host. If you do *not* use this argument, you will have to delete the existing trusted host with the Administrative UI before using the smreghost command. We recommend using the smreghost command with this argument.

The trusted host is re-registered.

Uninstall the SiteMinder WSS Agent

To uninstall the SiteMinder WSS Agent, run the CA SiteMinder® Web Services Security uninstall wizard.

Follow these steps:

1. Navigate to the *WSS_HOME*\install_config_info (Windows) or *WSS_HOME*/install_config_info (UNIX) directory and run the CA SiteMinder® Web Services Security uninstall wizard to remove CA SiteMinder® Web Services Security agents:

   - Windows: soa-uninstall.cmd
   - UNIX: soa-uninstall.sh

   WSS_HOME

   Specifies the CA SiteMinder® Web Services Security installation location.

   Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA SiteMinder® Web Services Security Release Notes.

   The uninstall wizard starts.

2. Choose whether you want to perform a complete uninstall or whether to uninstall specific features and proceed.
3. If you chose to uninstall only specific features, select the installed components that you want to uninstall and proceed.

   The uninstall wizard removes all selected CA SiteMinder® Web Services Security components.

4. Restart the server.

Chapter 5: Upgrade a SOA Agent to a 12.52 WSS Agent

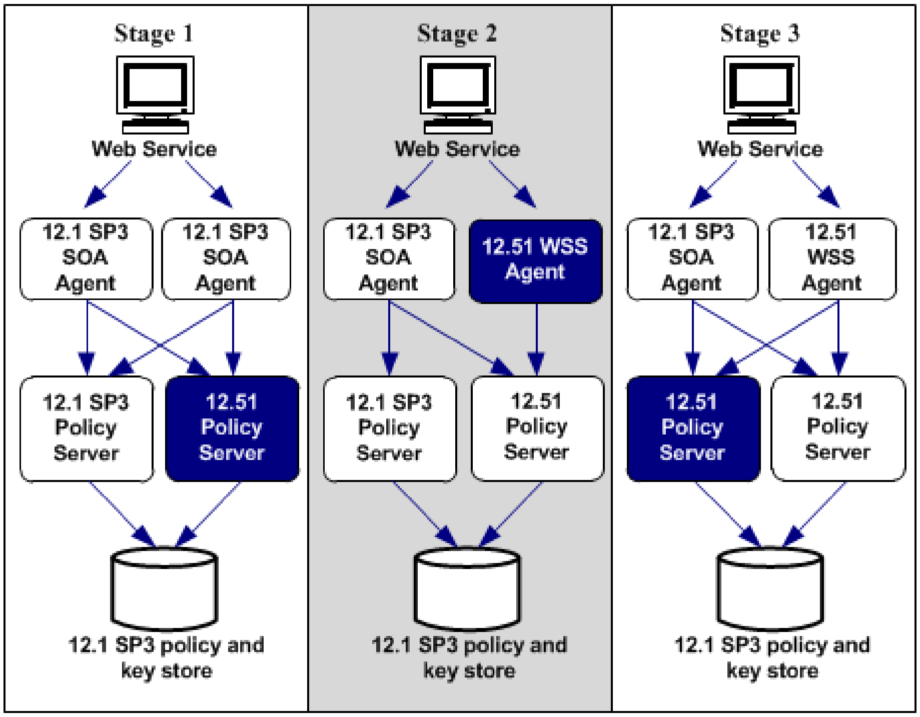
This section contains the following topics:

How to Upgrade a SOA Agent (see page 53)

How to Upgrade a SOA Agent

Upgrading a SOA Agent to a 12.52 WSS Agent involves several separate procedures. To upgrade your agent, Follow these steps::

1. Verify that you are in the proper step of the upgrade process for an agent upgrade. You upgrade agents to 12.52 from r12.1 SP3 at stage two of the CA SiteMinder® Web Services Security upgrade process, as shown in the following illustration:



2. Create backup copies of any customized agent-related files on your web server. Examples of files you could have customized after installing or configuring your agent include the following files:
   - LocalConfig.conf
   - WebAgent.conf
3. If you are upgrading an agent on a UNIX/Linux operating environment, clear the LD_PRELOAD variable (see page 54).
4. Gather information for the following CA SiteMinder® programs.
   - Agent installation wizard.
   - Agent configuration wizard.
5. Run the installation wizard to upgrade your agent on Windows (see page 55) or UNIX (see page 56).
6. If you are upgrading an agent on a UNIX/Linux operating environment, source the agent environment script on the upgraded agent).
7. Run the configuration wizard to configure the upgraded agent on Windows (see page 57) or UNIX (see page 58).

Verify That the LD_PRELOAD Variable Does Not Conflict with Existing Agent

If you are upgrading or reinstalling a SiteMinder WSS Agent on a Linux system, from the shell, set the LD_PRELOAD variable so that it points to a different location from any existing agent installation directory. For example, if an existing LD_PRELOAD entry is set to:

LD_PRELOAD=*agent_home*/bin/libbtunicode.so

Before you reinstall or upgrade, set the variable to:

```
export LD_PRELOAD=
```

This entry sets the variable to a blank value.

Run the Installation Wizard to Upgrade Your Agent on Windows

The installation program for the SiteMinder WSS Agent installs the agent on one computer at a time using the Windows operating environment. This installation program can be run in wizard or console modes. The wizard and console-based installation programs also create a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

Follow these steps:

1. Copy the SiteMinder WSS Agent installation executable file to a temporary directory on your web server.
2. Do *one* of the following steps:
   - For wizard-based installations, right-click ca-sm-wss-<SVMVER>-*cr*-win32.exe, and then select Run as Administrator.

     cr

     Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.
   - For console-based installations, open a command line window and run the executable as shown in the following example:

     ```
     ca-sm-wss-<SVMVER>-cr-win32.exe -i console
     ```

   Important! If you are running this wizard on Windows Server 2008, run the executable file with Administrator permissions, even if you are logged into the system as an Administrator. For more information, see the CA SiteMinder® Web Services Security Release Notes.
3. Use the information that you gathered previously to complete the installation.

Note: The software upgrade occurs in the installed location of the existing SOA Agent.

Run the Installation Wizard to Upgrade your Agent on UNIX/Linux

The installation program for the SiteMinder WSS Agent installs the agent on one computer at a time using the UNIX or Linux operating environments. This installation program can be run in wizard or console modes. The wizard and console-based installation program also creates a .properties file for subsequent installations and configurations using the unattended or silent method with the same settings.

For example, suppose the Agents in your environment use the same web server version, installation directory, Agent Configuration Object and Policy Servers. Use the installation wizard or console-based installation program for your first installation. Afterwards, you could create your own script to run the installation program with the .properties file the wizard or console-based installation program created.

Follow these steps:

1. Copy the SiteMinder WSS Agent installation executable file to a temporary directory on your web server.
2. Log in as a root user.
3. Do *one* of the following steps:
   - For wizard-based installations, run ca-sm-wss-<SVMVER>-*cr*-*unix_version*.bin

     cr

     Specifies the cumulative release number. The base 12.52 release does not include a cumulative release number.

     unix_version

     Specifies the UNIX version: **sol** or **linux**...
   - For console-based installations, open a command-line window and run the executable as shown in the following example:

     ```
     ca-sm-wss-<SVMVER>-cr-unix_version.bin -i console
     ```
4. Use the information from your agent Installation worksheet to complete the installation program.

Note: The software upgrade occurs in the installed location of the existing SOA Agent.

Set the Library Path Variable Before Configuring your Upgraded Agent on UNIX/Linux

Set the library path variable on UNIX or Linux systems before running the agent configuration program.

The following table lists the library path variables for the various UNIX and Linux operating environments:

Operating System	**Name of Library Path Variable**
AIX	LIBPATH
Linux	LD_LIBRARY_PATH
Solaris	LD_LIBRARY_PATH

Set the value of the library path variable to the *web_agent_home*/bin directory.

web_agent_home

Indicates the directory where the CA SiteMinder® Agent is installed.

Default (UNIX/Linux installations): [set the Installation Path variable]/webagent

Run the Configuration Wizard on Your Upgraded SiteMinder WSS Agent on Windows

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

Follow these steps:

1. Open the following directory on your web server:

   ```
   WSS_Home\install_config_info
   ```

   WSS_Home

   Specifies the path to where CA SiteMinder® Web Services Security is installed.

   Default: C:\Program Files\CA\Web Services Security

2. Use *one* of the following configuration methods:

   - For a GUI-based configuration, right-click ca-pep-config.exe, and then select Run as Administrator:
   - For a console-based configuration, enter the following command from a Command Prompt window with Administrator privileges open to *WSS_Home*\install_config_info:

     ```
     ca-pep-config.exe -i console
     ```

     Important! If you are running this wizard on Windows Server 2008, run the executable file with administrator permissions. Use these permissions even if you are logged in to the system as an administrator. For more information, see the release notes for your CA SiteMinder® component.

3. Use the information you gathered earlier to complete the wizard.
4. The agent runtime instance is created for your web servers.

Run the Configuration Wizard on Your Upgraded SiteMinder WSS Agent on UNIX/Linux

After gathering the information for your agent configuration, run the agent configuration program. This program creates an agent runtime instance for the web servers running on your computer.

This configuration program is wizard or console based, depending on the option you select. Running the configuration program in the wizard or console mode once creates a properties file. Use the properties file to run unattended configurations on other computers with same operating environment in the future.

Follow these steps:

1. Open a Console Window with root privileges on your web server:
2. Navigate to the following location:

   ```
   WSS_Home/install_config_info
   ```

   WSS_Home

   Specifies the path to where CA SiteMinder® Web Services Security is installed.

3. Enter one of the following commands:

   GUI Mode: ./ca-pep-config.bin

   Console Mode: ./ca-pep-config.bin -i console

   The Configuration Wizard starts.

4. Use *one* of the following configuration methods:

   - For a GUI-based configuration, run ca-pep-config.bin.
   - For a console-based configuration, open a Command prompt window with root privileges and enter the following command:

     ```
     ca-pep-config.exe -i console
     ```

5. Use the information you gathered earlier to complete the wizard.

   The agent runtime instance is created for your web servers.

Chapter 6: Configure the SiteMinder WSS Agent

This section contains the following topics:

How to Configure the SiteMinder WSS Agent (see page 61)
SiteMinder WSS Agent for WebLogic Agent Configuration File (see page 62)
Agent Configuration Object (see page 64)
SiteMinder WSS Agent Configuration Parameters (see page 64)
Configure the Username and Password Digest Token Age Restriction (see page 68)

How to Configure the SiteMinder WSS Agent

To configure the SiteMinder WSS Agent, you must specify the following:

- Host Configuration Object (one for each host server)
- Agent Configuration Object (one for each SiteMinder WSS Agent)
- Agent identity (one for each SiteMinder WSS Agent)

Note: For detailed information about how to configure Agent-related objects, see *the CA SiteMinder® Policy Server Configuration Guide* and the *CA SiteMinder® Implementation Guide*.

Follow these steps:

1. On the Policy Server:

    a. Duplicate or create a Host Configuration Object, which holds initialization parameters for a Trusted Host.

       The Trusted Host is a server that hosts one or more Agents and handles their connection to the Policy Server.

    b. As necessary, add or edit parameters in the Host Configuration Object that you just created.

    c. Duplicate or create an Agent Configuration Object, which holds Agent configuration parameters and can be used to centrally configure a group of Agents.

    d. Add or edit required Agent parameters in the Agent Configuration Object.

       The configuration object must include the DefaultAgentName or AgentName parameter to specify the Agent identity.

    e. Create an Agent identity for the SiteMinder WSS Agent. You must select *Web Agent* as the Agent type for a SiteMinder WSS Agent.

2. On the system where the SiteMinder WSS Agent is installed:

   a. Run the Agent Configuration Wizard, which registers the Trusted Host.

   b. Enable the SiteMinder WSS Agent by setting the EnableWebAgent parameter in the Agent configuration file to Yes.

SiteMinder WSS Agent for WebLogic Agent Configuration File

By default, the SiteMinder WSS Agent for WebLogic installation creates a single agent configuration file, JavaAgent.conf. The agent configuration file is located in the *WSS_Home*/config directory.

WSS_Home

Specifies the location where the SiteMinder WSS Agent is installed.

Each agent configuration file is created with the following required default configuration parameters/values:

Parameter	Description
DefaultAgentName	The agent identity the Policy Server uses to associate policies with the SiteMinder WSS Agent. The default value is "SoaAgent". Do not change this value.
EnableAgent	Specifies whether the SiteMinder WSS Agent is enabled. Possible values are Yes and No. Default value is Yes.
AgentConfigObject	The Agent Configuration Object specified during installation.
SmHostFile	Path to the local Host Configuration File. Path can be specified in absolute terms or relative to *WSS_HOME.* **Note:** On Windows, specify paths using double backslashes ("\\") rather than single backslash ("\") to separate directories. On UNIX, use standard single slash ("/") separators. Example values: ■ (Windows) C:\\Program Files\\CA\\Web Services Security\\wlsagent\\config\\SmHost.conf ■ (UNIX) /config/SmHost.conf
ServerName	A string that will be used in the SiteMinder WSS Agent log to identify the WebLogic Server. Default value is "SOAWLS92".

Parameter	Description
appserverjaasloginhandler	Specifies the Application Server-specific SiteMinder WSS Agent handler class for WebLogic Default value is "com.ca.soa.agent.appserver.jaas.wls.WlsLoginHandler". Do not change this value.

You need only edit the preconfigured values if the location of the Host Configuration File changes or you want to refer to a different Agent Configuration Object. If you use local configuration, you can add other Agent configuration parameters to these preconfigured values.

Note: Parameters that are held in the Agent configuration file are static. If you change these settings while the WebLogic server is running, the SiteMinder WSS Agent does not pick up the change until WebLogic is restarted.

The JavaAgent.conf file also contains a list of SiteMinder WSS Agent plugin classes; you do not need to alter this information.

Note: Leading and trailing whitespace in JavaAgent.conf value definitions is ignored. To include leading or trailing whitespace, quote the value (with either single or double quotes). Embedded, escaped quotes are unescaped during processing.

Sample JavaAgent.conf (Windows)

```
# Java Agent Configuration File
#
# This file contains bootstrap information required by
# the SiteMinder Java Agent
#
#
# Configuration for agent testagent
#
defaultagentname=soaagent
enablewebagent=yes
agentconfigobject=soaagentconfig
servername=wsdell110.systemtest.local
smhostfile=C:\\SOASecurityManager\\wlsagent\\config\\SmHost.conf
appserverjaasloginhandler=com.ca.soa.agent.appserver.jaas.wls.WLSLoginHandler
appserverjmshandler=com.ca.soa.agent.appserver.jaxrpc.jms.wls.WLSJMSMessageHandle
r
# Configure plugins for the agent testagent
transport_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig,
com.ca.soa.agent.jaxrpcplugin.pluginconfig.JaxRpcPluginConfig,
com.ca.soa.agent.jmsplugin.pluginconfig.JMSPluginConfig
msg_body_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
credential_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig,
com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig
variable_resolver_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginCo
nfig
# <EOF>
```

Agent Configuration Object

An Agent Configuration Object is a CA SiteMinder® policy object that holds Agent parameters for an Agent when using central agent configuration.

Note: Parameters held in an Agent Configuration Object are dynamic; if you change these settings while the WebLogic server is running, the SiteMinder WSS Agent will pick up the change.

SiteMinder WSS Agent Configuration Parameters

The following table contains a complete list of all Agent configuration parameters supported by SiteMinder WSS Agents for Application Servers.

Unless otherwise noted,you can define parameters in either the Agent Configuration Object or the Agent configuration file depending upon how you decide to configure the SiteMinder WSS Agent.

Parameter Name	Value	Description
AcceptTPCookie	YES or NO	(Optional) If set to yes, configures the SiteMinder WSS Agent to assert identities from third-party SiteMinder session cookies (that is, session cookies generated by custom Agents created using the SiteMinder and SiteMinder WSS SDKs. **Note:** AcceptTPCookie must be set to Yes to assert identities from session cookies generated by CA SOA Security Gateway. Default is Yes.
AllowLocalConfig (Applies only in the Agent Configuration Object)	YES or NO	If set to yes, parameters set locally in the Agent configuration file take precedence over parameters in the Agent Configuration Object. Default is NO.
AuthCacheSize	Number	(Optional) Size of the authentication cache for the SiteMinder WSS Agent (in number of entries). For example: `authcachesize="1000"` Default is 0. To flush this cache, use the Policy Server User Interface.
AzCacheSize		(Optional) Size of the authorization cache (in number of entries) for the SiteMinder WSS Agent. For example: `authcachesize="1000"` Default is 0. To flush this cache, use the Policy Server User Interface.
CacheTimeout	Number	(Optional) Number of seconds before cache times out. For example: `cachetimeout="1000"` Default is 600 (10 minutes).
ConfigObject (Applies only in Agent configuration file)	String	The name of the Agent Configuration Object associated with the SiteMinder WSS Agent. No default value.

Parameter Name	Value	Description
CookieDomain	String	(Optional) Name of the cookie domain. For example: `cookiedomain="ca.com"` No default value. For more information, see the cookiedomainscope parameter.
CookieDomainScope	Number	(Optional) Further defines the cookie domain for assertion of SiteMinder session cookies by the SiteMinder WSS Agent. The scope determines the number of sections, separated by periods, that make up the domain name. A domain always begins with a period (.) character. For example: `cookiedomainscope="2"` Default is 0, which takes the domain name specified in the cookiedomain parameter.
DefaultAgentName (Applies only in the Agent Configuration Object)	String	The agent identity the Policy Server will use to associate policies with the SiteMinder WSS Agent. Default is "SoaAgent"; this value should not changed.
EnableWebAgent (Applies only in Agent configuration file)	YES or NO	Enables or disables the SiteMinder WSS Agent. When set to 'yes', the SiteMinder WSS Agent will protect resources using the Policies configured in the Policy Server for the configured agent identity. Default is Yes.
LogOffUri	String	(Optional) The URI of a custom HTTP file that will perform a full log off (removing the session cookie from a user’s browser). A fully qualified URI is not required. For example, LogOffUri could be set to: /Web pages/logoff.html No default value.
PsPollInterval	Number	(Optional) The frequency with which the SiteMinder WSS Agent polls the Policy Server to retrieve information about policy changes. Default is 30 seconds.

Parameter Name	Value	Description
ResourceCacheSize	Number	(Optional) Size (in number of entries) of the cache for resource protection decisions. For example: `resourcecachesize="1000"` Default is 2000. To flush this cache, use the Policy Server User Interface.
SAMLSessionTicketLogoffi	YES or NO	(Optional) Determines whether the SiteMinder WSS Agent should attempt to log off session tickets in SAML assertions. Default is Yes.
ServerName (Applies only in Agent configuration file.)	String	A string to be used in the SiteMinder WSS Agent log to identify the target application server.
SessionGracePeriod	Number	(Optional) Grace period (in seconds) between the regeneration of session tokens. Default is 30
SmHostFile (Applies only in Agent configuration file)	String	Path to the local Host Configuration File (typically *WSS_Home*\conf\SmHost.conf). No default value.
XMLAgentSoapFaultDetails	YES or NO	(Optional) Determines whether or not the SiteMinder WSS Agent should insert the authentication/authorization rejection reason (if provided by the Policy Server) into the SOAP fault response sent to the web service consumer. Default is No.
XMLSDKAcceptSMSessionCookie	YES or NO	(Optional) Determines whether or not the SiteMinder WSS Agent accepts an CA SiteMinder session cookie to authenticate a client. Default is No. If set to Yes, the SiteMinder WSS Agent uses information in a session cookie sent as an HTTP header in the request as a means of authenticating the client. If set to No, session cookies are ignored and the SiteMinder WSS Agent requests credentials required by the configured authentication scheme.

Parameter Name	Value	Description
XMLSDKMimeTypes	String	(Optional) A comma-delimited list of MIME types that the SiteMinder WSS Agent will accept for processing by CA SiteMinder® Web Services Security. All POSTed requests having one of the listed MIME types are processed. Examples: ■ text/xml ■ application/octet-stream ■ text/xml,multipart/related If you do not add this parameter to the Agent Configuration Object, the SiteMinder WSS Agent defaults to accepting text/xml and application/soap+xml MIME types.

Configure the Username and Password Digest Token Age Restriction

By default, the WS-Security authentication scheme imposes a 60-minute restriction on the age of Username and Password Digest Tokens to protect against replay attacks.

To configure a different value for the token age restriction for a SiteMinder WSS Agent for Application Servers, set the WS_UT_CREATION_EXPIRATION_MINUTES parameter in the XmlToolkit.properties file for that agent.

Follow these steps:

1. Navigate to *WSS_Home*\wlsagent\config.
2. Open XmlToolkit.properties in a text editor.
3. Uncomment and modify the WS_UT_CREATION_EXPIRATION_MINUTES parameter line to configure a different value for the token age restriction:

   ```
   WS_UT_CREATION_EXPIRATION_MINUTES=token_age_limit
   ```

   token_age_limit

   Specifies the token age limit restriction in minutes.

4. Save and close the XmlToolkit.properties file.
5. Restart the SiteMinder WSS Agent.

Chapter 7: Set the WebLogic Environment for the SiteMinder WSS Agent

This section contains the following topics:

WebLogic Environment Setting Locations (see page 69)
Set the WebLogic Environment on Windows (see page 70)
Set the WebLogic Environment on UNIX (see page 71)

WebLogic Environment Setting Locations

You configure SiteMinder WSS Agent-related environment settings in one of the following locations depending on your environment:

- The WebLogic start script for both managed and standalone servers (startWebLogic.cmd on Windows; startWebLogic.sh on UNIX)

  Note: The startWebLogic.cmd (Windows) or startWebLogic.sh (Unix) script that contains the environment configuration is placed in the bin folder of a created domain.

- If using the Node Manager to control Managed Servers, in the Server Start configuration page in the WebLogic Adminstration Console.

  For details regarding the Server Start configuration page, see the WebLogic Online Documentation.

Set the WebLogic Environment on Windows

Before the SiteMinder WSS Agent can operate with the WebLogic Application Server on Windows, you must configure SiteMinder WSS Agent-related environment settings in the location appropriate for your environment.

Follow these steps:

1. Define a SOA_HOME variable as follows:

   ```
   set SOA_HOME=WSS_Home\wlsagent
   ```

2. Define a SMSOA_CLASSPATH variable as follows:

   ```
   set SMSOA_CLASSPATH=%SOA_HOME%\config;
   %SOA_HOME%\lib\smagentapi.jar;
   %SOA_HOME%\lib\thirdparty\cryptojFIPS.jar;
   %SOA_HOME%\lib\soaagent-proxy.jar;
   %SOA_HOME%\lib\thirdparty\xalan.jar
   ```

3. Add %SMSOA_CLASSPATH% to the beginning of the CLASSPATH variable definition. The modified CLASSPATH variable should resemble the following:

   ```
   set CLASSPATH=%SMSOA_CLASSPATH%;%CLASSPATH%
   ```

4. Define the SM_JAVA_OPTIONS variable as follows:

   ```
   set SM_JAVA_OPTIONS=-DJAVA_AGENT_ROOT=%SOA_HOME%
   -Dlog.log-config-properties=%SOA_HOME%\config\log-config.properties
   -Djava.security.auth.login.config==%SOA_HOME%\config\soa_jaas.config
   -Djavax.xml.soap.SOAPFactory=weblogic.xml.saaj.SOAPFactoryImpl
   -Djavax.xml.soap.MessageFactory=weblogic.xml.saaj.MessageFactoryImpl
   ```

5. Add %SM_JAVA_OPTIONS% to the execution entry. The modified execution entry should resemble the following:

   ```
   %JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
   %SM_JAVA_OPTIONS% -Dweblogic.Name=%SERVER_NAME%
   -Djava.security.policy=%WL_HOME%\server\lib\weblogic.policy %PROXY_SETTINGS%
   %SERVER_CLASS%
   ```

6. Save your changes.

7. Restart the WebLogic Application Server for changes to take effect.

More information:

WebLogic Environment Setting Locations (see page 69)

Set the WebLogic Environment on UNIX

Before the SiteMinder WSS Agent can operate with the WebLogic Application Server on UNIX, you must configure SiteMinder WSS Agent-related environment settings in the location appropriate for your environment.

Follow these steps:

1. Define a SOA_HOME variable as follows:

   ```
   SOA_HOME=WSS_Home/wlsagent
   ```

2. Define the SMSOA_CLASSPATH as follows:

   ```
   SMSOA_CLASSPATH=${SOA_HOME}/config:
   ${SOA_HOME}/lib/smagentapi.jar:
   ${SOA_HOME}/lib/thirdparty/cryptojFIPS.jar:
   ${SOA_HOME}/lib/soaagent-proxy.jar:
   ${SOA_HOME}/lib/thirdparty/xalan.jar
   ```

   ```
   export SMSOA_CLASSPATH
   ```

3. Add ${SMSOA_CLASSPATH} to the beginning of the CLASSPATH definition. The modified CLASSPATH variable should resemble the following:

   ```
   CLASSPATH=${SMSOA_CLASSPATH}${CLASSPATHSEP}${CLASSPATH}
   export CLASSPATH
   ```

4. Define the SM_JAVA_OPTIONS variable as follows:

   ```
   SM_JAVA_OPTIONS="-DJAVA_AGENT_ROOT=${SOA_HOME}
   -Dlog.log-config-properties=${SOA_HOME}/config/log-config.properties
   -Djava.security.auth.login.config==${SOA_HOME}/config/soa_jaas.config
   -Djavax.xml.soap.SOAPFactory=weblogic.xml.saaj.SOAPFactoryImpl
   -Djavax.xml.soap.MessageFactory=weblogic.xml.saaj.MessageFactoryImpl"
   ```

5. Add ${SM_JAVA_OPTIONS} to the execution entry. The modified execution entry should resemble the following:

   ```
   "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}
   ${SM_JAVA_OPTIONS} -Dweblogic.Name=${SERVER_NAME}
   -Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy
   ${PROXY_SETTINGS} ${SERVER_CLASS}"
   ```

6. Save your changes.

7. Restart the WebLogic Application Server for changes to take effect.

More information:

WebLogic Environment Setting Locations (see page 69)

Chapter 8: SiteMinder WSS Agent for Oracle WebLogic Logging

This section contains the following topics:

SiteMinder WSS Agent Logging (see page 73)
Log Files (see page 73)
Change the SiteMinder WSS Agent Log File Name (see page 74)
Append Messages to an Existing SiteMinder WSS Agent Log File (see page 75)
Set the SiteMinder WSS Agent File Log Level (see page 75)
Roll Over the SiteMinder WSS Agent Log File (see page 75)
Disable SiteMinder WSS Agent XML Message Processing Logging (see page 76)
SiteMinder WSS Agent Log Configuration File Summary (see page 76)
How to Set Log Files, and Command-line Help to Another Language (see page 77)

SiteMinder WSS Agent Logging

The SiteMinder WSS Agent logger for application servers is implemented using Apache's log4j. For more information, see http://logging.apache.org/log4j/docs/.

Log Files

Two log files provide important information about the SiteMinder WSS Agent:

- SiteMinder WSS Agent log file—Logs SiteMinder WSS Agent error and processing messages.
- SiteMinder WSS Agent XML message processing log file—Logs messages information relating specifically to the SiteMinder WSS Agent's processing of XML messages

SiteMinder WSS Agent Log

The SiteMinder WSS Agent for Oracle WebLogic writes information about its standard operations and performance to the SiteMinder WSS Agent log.

By default, SiteMinder WSS Agent logging is enabled and written to the XmlAgent.log file in:

- Windows—*WLS_HOME*\user_projects\domain\defaultdomain\soa-log
- UNIX—*WLS_HOME*/user_projects/domain/defaultdomain/soa-log

You can change SiteMinder WSS Agent logging parameters by editing the log-config.properties file located in:

- Windows—*WSS_Home*\wlsagent\config\
- UNIX— *WSS_Home*/wlsagent/config/

Note: These are the default values; the logging configuration file name and location can be changed by editing the log-config-properties JVM system property.

SiteMinder WSS Agent XML Message Processing Logging

In addition to its standard logging functionality, SiteMinder WSS Agents for Application Servers also logs information relating specifically to its processing of XML messages. Like the SiteMinder WSS Agent log, the XML message processing log is also implemented using Apache's *log4j* standard.

Note: SiteMinder WSS Agent XML message processing logging does not start until an XML message that needs to be processed is received.

By default, SiteMinder WSS Agent XML message processing logging is enabled and written to the soasm_agent.log file in:

- Windows—*WSS_Home*\wlsagent\bin\
- UNIX—*WSS_Home*/wlsagent/bin/

You can change SiteMinder WSS Agent XML message processing logging parameters by editing the log.config file, which can be found in:

- Windows—*WSS_Home*\wlsagent\config\
- UNIX— *WSS_Home*/wlsagent/config/

Change the SiteMinder WSS Agent Log File Name

To change pathname of the SiteMinder WSS Agent log file, edit the log.logfile-pattern parameter. Possible values are valid pathnames. If you specify a relative value, the path is set relative to the JAVA_AGENT_ROOT JVM system property.

Default value: "log\XmlAgent.log"

For example:

```
log.logfile-pattern=log\XmlAgent.log
```

Append Messages to an Existing SiteMinder WSS Agent Log File

To add logging information to an existing SiteMinder WSS Agent log file instead of rewriting the entire file each time logging is invoked, add the log.logfile-append-on-reset parameter.

For example:

```
log.logfile-append-on-reset=YES
```

Set the SiteMinder WSS Agent File Log Level

To change the SiteMinder WSS Agent log level, edit the log.logging-level parameter. Possible values are:

- DEBUG - Logs all; most verbose
- CONFIG - Configuration information
- INFO - Information
- WARNING - Warnings
- SEVERE - Errors only; least verbose

Default value: WARNING

For example:

```
log.logging-level=INFO
```

Roll Over the SiteMinder WSS Agent Log File

To change file size limit at which the SiteMinder WSS Agent log should rollover, change the log.logfile-limit parameter. Rolling over a log file starts a new log file, preventing a single log file from becoming unmanageable. Possible values are numbers, representing kilobytes.

The default value is 1000.

For example:

```
log.logfile-limit=512
```

Disable SiteMinder WSS Agent XML Message Processing Logging

To disable SiteMinder WSS Agent XML message processing logging, remove or comment out (using a "#" prefix) the following lines from the log.config file located in the Agent config subdirectory:

```
log4j.appender.A2=org.apache.log4j.DailyRollingFileAppender
log4j.appender.A2.File=${NETE_TXM_ROOT}/bin/soasm_agent.log
```

SiteMinder WSS Agent Log Configuration File Summary

The SiteMinder WSS Agent logging configuration file defines default SiteMinder WSS Agent logging settings.

Available configuration parameters are:

Name	Description
log.logfile-append-on-reset	Add logging information to an existing log file instead of creating a new file each time logging is invoked. Default value: no
log.logfile-pattern	Specifies the pathname (relative to *WSS_Home*) of the SiteMinder WSS Agent log file. Default value: log/XmlAgent.log
log.logging-level	Defines the logging level. The levels are: ■ DEBUG - all logging, most verbose ■ CONFIG - configuration information ■ INFO - information ■ WARNING -warnings ■ SEVERE - errors Default value: WARNING
log.logfile-limit	Specifies the size limit, in kilobytes Rollover a log file after it reaches the specified size. Default value: 1,000KB

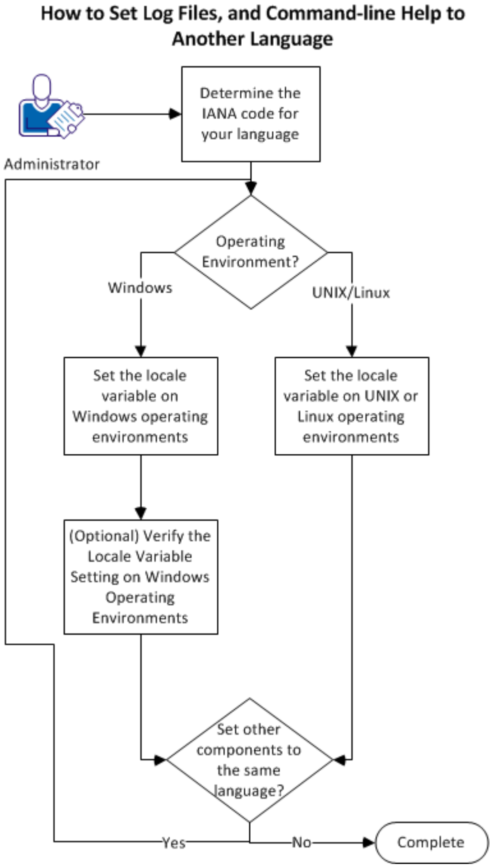
Note: Once the SiteMinder WSS Agent connects to the Policy Server, corresponding logging settings found in the Agent Configuration Object override the values in log-config.properties.

How to Set Log Files, and Command-line Help to Another Language

The following components support log files, and command-line help in other languages:

- The Policy Server
- The Web Agent
- The Report Server
- The CA SiteMinder Agent for SharePoint
- The CA SiteMinder® SPS
- SiteMinder WSS Agents
- Any custom software that is created with the CA SiteMinder® SDK.

The following graphic describes the work flow for setting log files, and command-line help to another language:


How to Set Log Files, and Command-line Help to Another Language
Administrator
Determine the IANA code for your language
Operating Environment?
Windows
UNIX/Linux
Set the locale variable on Windows operating environments
Set the locale variable on UNIX or Linux operating environments
(Optional) Verify the Locale Variable Setting on Windows Operating Environments
Set other components to the same language?
Yes
No
Complete

Follow these steps:

1. Determine the IANA code for your language (see page 78).
2. Create the environment variable for your operating environment using one of the following procedures:
   - Set the locale variable on Windows operating environments (see page 80).
   - Set the locale variable on UNIX or Linux operating environments (see page 82).
3. (Optional) Verify the locale variable setting on windows operating environments (see page 81).
4. (Optional) Repeat Steps 1 through 3 to set any other components in your environment to the same language.

Determine the IANA Code for Your Language

Each language has a unique code. The Internet Assigned Numbers Authority (IANA) assigns these language codes. Adding a language code to a locale variable changes the language that the software displays. Determine the proper code for the language that you want before creating the locale variable.

The following table lists the IANA codes that correspond to the languages supported by the software:

Language	**IANA Code**
Brazilian Portuguese	pt_BR
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Simplified Chinese	zh-Hans
Spanish	es

Note: A list of IANA language codes is available from this third-party website.

Environment Variables

The environment variables are settings by which users can customize a computer to suit their needs. Examples of environment variables include the following items:

- A default directory for searching or storing downloaded files.
- A username.
- A list of locations to search for executable files (path).

Windows operating environments allow global environment variables, which apply to all users of a computer. The environment variables on UNIX or Linux operating environments must be set for each user or program.

To set the locale variable, pick the procedure for your operating environment from the following list:

- Set the locale variable on Windows operating environments (see page 80).
- Set the locale variable on UNIX or Linux operating environments (see page 82).

Set the Locale Variable on Windows Operating Environments

The following locale variable specifies the language settings for the software:

```
SM_ADMIN_LOCALE
```

Create this variable and set it to the language that you want. Set this variable on *each* component for which you want to use another language. For example, suppose you want to have a Policy Server and an agent that is set to French. Set this variable on both of those components to French.

Note: The installation or configuration programs do *not* set this variable.

Follow these steps:

1. Click Start, Control Panel, System, Advanced system settings.

   The system properties dialog appears.

2. Click the Advanced tab.
3. Click Environment Variables.
4. Locate the System variables section, and then click New.

   The New System Variable dialog opens with the cursor in the Variable name: field.

5. Type the following text:

   ```
   SM_ADMIN_LOCALE
   ```

6. Click the Variable name: field, and then type the IANA language code (see page 78) that you want.
7. Click OK.

   The New System Variable dialog closes and the SM_ADMIN_LOCALE variable appears in the list.

8. Click OK *twice*.

   The locale variable is set.

9. (Optional) Repeat Steps 1 through 8 to set other components to the same language.

Verify the Locale Variable Value on Windows Operating Environments

You can very the value to which the locale variable is set at any time. You can do this procedure after setting the variable to confirm that it is set correctly.

Note: Instructions for verifying the variable value on UNIX and Linux are in the setting procedure (see page 82).

Follow these steps:

1. Open a command-line window with the following steps:

   a. Click Start, Run.

   b. Type the following command:

      ```
      cmd
      ```

   c. Click OK.

      A command-line window opens.

2. Enter the following command:

   ```
   echo %SM_ADMIN_LOCALE%
   ```

   The locale appears on the next line. For example, when the language is set to German, the following code appears:

   ```
   de
   ```

   The value of the locale variable is verified.

Set the Locale Variable on UNIX or Linux Operating Environments

The following locale variable specifies the language settings for the software:

```
SM_ADMIN_LOCALE
```

Create this variable and set it to the language that you want. Set this variable on *each* component for which you want to use another language. For example, suppose you want to have a Policy Server and an agent that is set to French. Set this variable on both of those components to French.

Note: The installation or configuration programs do *not* set this variable.

Follow these steps:

1. Log in to the computer that is running the component that you want.
2. Open a console (command-line) window.
3. Enter the following command:

   ```
   export SM_ADMIN_LOCALE=IANA_language_code
   ```

   The command in the following example sets the language to French:

   ```
   export SM_ADMIN_LOCALE=fr
   ```

   The locale variable is set.
4. (Optional) Verify that the locale variable is set properly by entering the following command:

   ```
   echo $SM_ADMIN_LOCALE
   ```

   The locale appears on the next line. For example, when the language is set to German, the following code appears:

   ```
   de
   ```
5. (Optional) Repeat Steps 1 through 4 to set other components to the same language.

Chapter 9: Final Steps

This section contains the following topics:

Prevent WebLogic 10 from Loading Incompatible Version of XML Security (see page 83)
Restart WebLogic (see page 83)
Configure Web Services to Invoke the SiteMinder WSS Agent JAX-RPC Handler (see page 83)
Configure Policies for the SiteMinder WSS Agent (see page 87)
Troubleshoot Issues Related to Protected Web Services Making Use of Apache Commons Logging and Log4j (see page 87)

Prevent WebLogic 10 from Loading Incompatible Version of XML Security

By default, Weblogic Server 10 loads an older version of XML security (1.3.0) that is incompatible with the version used by the SiteMinder WSS Agent (1.4.1).

To prevent WebLogic 10 from loading the 1.3.0 XML security JAR, rename *WLS_HOME*\modules\com.bea.core.apache.xml.security_1.3.0.jar to some other name. For example, com.bea.core.apache.xml.security_1.3.0_backup.jar.

Restart WebLogic

After completing WebLogic-side configuration of the SiteMinder WSS Agent, you must restart the WebLogic server.

Configure Web Services to Invoke the SiteMinder WSS Agent JAX-RPC Handler

To protect a JAX-RPC web service using the SiteMinder WSS Agent, you must configure it to invoke the SiteMinder WSS Agent JAX-RPC Handler. To do this, you must add the SiteMinder WSS Agent JAX-RPC Handler class (com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandlerProxy) to the web service deployment descriptor in the webservices.xml file.

You can do this manually by editing the webservices.xml file for each JAX-RPC web service module. However, if your web services are implemented as JWS files and you have set up an Ant-based development environment, it is more efficient to update your web services to use handler chains.

Manually Edit JAX-RPC Web Service Deployment Descriptors

To configure JAX-RPC web services not implemented as JWS files to invoke the SiteMinder WSS Agent, you must manually edit their deployment descriptors to add the SiteMinder WSS Agent JAX-RPC Handler.

Follow these steps:

1. Unpack the enterprise archive (EAR) containing one or more web services.
2. Examine the EAR to determine which of the modules within it contains a JAX-RPC web service.(A module that contains a JAX-RPC web service if it has a webservices.xml file in the META-INF folder for EJB endpoints, or the WEB-INF folder for servlet endpoints.)
3. For each module in the EAR identified as a JAX-RPC web service:
    a. Unpack the archive containing the module. (The archive will be a JAR file for EJB endpoints and a WAR file for servlet endpoints.)
    b. Find the webservices.xml file.
    c. For each port-component element found in the webservices.xml file, add a handler element:

    ```
    <handler>
       <handler-name>WSS Agent Handler<handler-name>
       <handler-class>
          com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandlerProxy
       </handler-class>
    </handler>
    ```

    Note: The WSS Agent JAX-RPC handler must always be invoked first; If other handler elements are already present or subsequently added to the webservices.xml file, the WSS Agent JAX-RPC Handler element must be placed before them.
4. Repackage the module into the appropriate archive type (JAR or WAR).
5. When all modules have been configured, repackage the EAR.
6. Install or update the enterprise application.

Use Handler Chains

The most efficient way to configure services implemented as JWS files to invoke the SiteMinder WSS Agent is to define the SiteMinder WSS Agent JAX-RPC Handler class in a handler chain configuration file which can then be referenced from the JWS files of all web services in your enterprise that you need to protect.

Note: The following procedures assume that you have set up an Ant-based development environment and have a working build.xml file that includes a target for running the jwsc Ant task.

Use a Handler Chain to Invoke the SiteMinder WSS Agent for HTTP Requests

To configure services implemented as JWS files to invoke the SiteMinder WSS Agent for requests received over HTTP transport, define the SiteMinder WSS Agent JAX-RPC Handler class in a handler chain configuration file.

Follow these steps:

1. Create a handler chain configuration file that defines a JAX-RPC handler chain. The chain can include as many handler classes as you require but must define the SiteMinder WSS Agent JAX-RPC Handler class first.

   Example HandlerConfig.xml:

   ```
   <jwshc:handler-config xmlns:jwshc="http://www.bea.com/xml/ns/jws"
    xmlns:soap1="http://HandlerInfo.org/Server1"
    xmlns:soap2="http://HandlerInfo.org/Server2"
    xmlns="http://java.sun.com/xml/ns/j2ee" >
     <jwshc:handler-chain>
      <jwshc:handler-chain-name>HandlerChainName</jwshc:handler-chain-name>
      <jwshc:handler>
         <handler-name>handlerOne</handler-name>

   <handler-class>com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandlerProxy
         </handler-class>
       </jwshc:handler>
     </jwshc:handler-chain>
   </jwshc:handler-config>
   ```

   HandlerChainName

   Specifies the name of handler chain.

2. Add the JWS annotation @HandlerChain(file="HandlerConfig.xml", name="HandlerChainName") to the web service JWS file.
3. Rebuild the JWS web service.

WebLogic server will invoke SiteMinder WSS Agent JAX-RPC handler.

Note: For more information on SOAP message handlers and handler chains, see the WebLogic documentation.

Use a Handler Chain to Invoke the SiteMinder WSS Agent for JMS Requests

To configure services implemented as JWS files to invoke the SiteMinder WSS Agent for requests received over JMS transport, define the SiteMinder WSS Agent JAX-RPC Handler class in a handler chain configuration file.

Follow these steps:

1. Create a handler chain configuration file that defines a JAX-RPC handler chain. The chain can include as many handler classes as you require but must define the SiteMinder WSS Agent JAX-RPC Handler class first.

   Example HandlerConfig.xml:

```
<jwshc:handler-config xmlns:jwshc="http://www.bea.com/xml/ns/jws"
 xmlns:soap1="http://HandlerInfo.org/Server1"
 xmlns:soap2="http://HandlerInfo.org/Server2"
 xmlns="http://java.sun.com/xml/ns/j2ee" >
  <jwshc:handler-chain>
   <jwshc:handler-chain-name>HandlerChainName</jwshc:handler-chain-name>
   <jwshc:handler>
      <handler-name>handlerOne</handler-name>
      <handler-class>
        com.ca.soa.agent.appserver.jaxrpc.jms.XMLAgentJMSJaxrpcHandlerProxy
      </handler-class>
    </jwshc:handler>
  </jwshc:handler-chain>
</jwshc:handler-config>
```

   HandlerChainName

   Specifies the name of handler chain.

2. Add the JWS annotation @HandlerChain(file="HandlerConfig.xml", name="HandlerChainName") to the web service JWS file.
3. Rebuild the JWS web service.

WebLogic server will invoke SiteMinder WSS Agent JAX-RPC handler for JMS requests.

Note: For more information on SOAP message handlers and handler chains, see the WebLogic documentation.

Configure Policies for the SiteMinder WSS Agent

You create authentication and authorization policies to protect web service resources hosted on WebLogic from their associated WSDL files using the Administrative UI. For more information, see the *CA SiteMinder® Web Services Security Policy Configuration Guide*.

Troubleshoot Issues Related to Protected Web Services Making Use of Apache Commons Logging and Log4j

Because the SiteMinder WSS Agent for WebLogic implements logging using the Apache Commons Logging and Apache log4j packages, class loading problems can occur if protected web service applications also use the Commons Logging and log4j packages.

Commons Logging Class Loading Problem

Symptom:

The SiteMinder WSS Agent for WebLogic uses the Apache Commons Logging package as part of its logging implementation. Class loading conflicts can occur when the SiteMinder WSS Agent is configured to protect web services that also use Commons Logging if the web service and SiteMinder WSS Agent are not using the same version of Commons Logging.

If such a class loading conflict is occurring, WebLogic may log an error output similar to the following during initialization:

```
<Sep 16, 2009 9:15:54 AM EDT> <Error> <HTTP> <BEA-101216> <Servlet:
"ServicesServlethttp"
failed to preload on startup in Web application: "XXXXXX.war".
java.lang.ExceptionInInitializerError
….
org.apache.commons.logging.LogConfigurationException:
org.apache.commons.logging.LogConfig
urationException: org.apache.commons.logging.LogConfigurationException: Invalid
class loader hierarchy.  You have more than one version of
'org.apache.commons.logging.Log' visible, which is not allowed. (Caused by
org.apache.commons.logging.LogConfigurationException: Invalid class loader
hierarchy.  You have more than one version of 'org.apache.commons.logging.Log'
visible, which is not allowed.) (Caused by
org.apache.commons.logging.LogConfigurationException:
org.apache.commons.logging.LogConfigurationException: Invalid class loader
hierarchy.  You have more than one version of 'org.apache.commons.logging.Log'
visible, which is not allowed. (Caused by
org.apache.commons.logging.LogConfigurationException: Invalid class loader
hierarchy.  You have more than one version of 'org.apache.commons.logging.Log'
visible, which is not allowed.))
        at
org.apache.commons.logging.impl.LogFactoryImpl.newInstance(LogFactoryImpl.java:54
3)
….
```

Solution:

To remove class loading conflicts between the SiteMinder WSS Agent and protected web service, verify that the web service uses the same version of the Apache Commons Logging jar file (commons-logging.jar) as the SiteMinder WSS Agent (version 1.1.1). If the web service is not using the same version, replace the commons-logging.jar used by the web service with the 1.1.1 version.

log4j Conflict Problems

Symptom:

The SiteMinder WSS Agent for WebLogic uses the Apache log4j package to produce logging output. This can lead to class loading and logging output conflict problems when the SiteMinder WSS Agent is configured to protect web services that also use the log4j package.

If such a conflict is occurring WebLogic logs error output similar to the following:

```
log4j:ERROR A "org.apache.log4j.DailyRollingFileAppender" object is not assignable
to a "o
rg.apache.log4j.Appender" variable.
log4j:ERROR The class "org.apache.log4j.Appender" was loaded by
log4j:ERROR [java.net.URLClassLoader@d8116d] whereas object of type
log4j:ERROR "org.apache.log4j.DailyRollingFileAppender" was loaded by
[weblogic.utils.clas
sloaders.ChangeAwareClassLoader@10abd60 finder:
weblogic.utils.classloaders.CodeGenClassFinder@5e33d4 annotation:
StockTradeDemoDCC@StockTradeDemoDCC.war].
log4j:ERROR Could not instantiate appender named "CUSTOMER_APPENDER_NAME".
```

This error is caused by the web service class loader loading the specific Appender implementation class: DailyRollingFileAppender and the SiteMinder WSS Agent class loader loading the Appender interface class.

Solution:

Configure the system class loader to load the log4j jar instead of the SiteMinder WSS Agent and web service class loaders by adding log4j.jar to the JVM classpath.

Follow these steps:

1. Add *WLS_HOME*/weblogic*XY*/server/lib/log4j.jar to the classpath variable passed into the Java Virtual Machine at startup.

   For a WebLogic Server running as a command line application, make this change by editing the startWeblogic.cmd (Windows) or startWeblogic.sh (UNIX) script located in *WLS_HOME*/user_projects/domains/*your_domain*/bin.

2. Rename the SiteMinder WSS Agent log4j.jar to prevent it from being loaded:

   a. Navigate to *WSS_Home*/wlsagent/lib/thirdparty.

   b. Rename log4j.jar to log4j.jar_hidden.

3. For each protected web service that also uses the log4j package, rename log4j.jar to prevent it from being loaded:

   a. Navigate to the directory that contains the log4j.jar file used by the web service.

   b. Rename log4j.jar to log4j.jar_hidden.

4. Modify the XML message processing logging configuration:

   a. Navigate to *WSS_Home*/wlsagent/config

   b. Open the log.config file in text editor.

   c. Locate the line that defines the log4j.category.com.netegrity.tm parameter. For example:

      ```
      log4j.category.com.netegrity.tm=DEBUG
      ```

   d. Append an explicit reference to the Message Processing Appender also declared in the log.config file. For example, if the name of the Daily Rolling File Appender is "A2", the log4j.category.com.netegrity.tm parameter definition line becomes:

      ```
      log4j.category.com.netegrity.tm=DEBUG, A2
      ```