CA SiteMinder® Agent for Oracle WebLogic Server

Agent Guide

r12.0 SP2

ca technologies

This documentation and any related computer software help programs (hereinafter referred to as the "Documentation") are for your informational purposes only and are subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be used or disclosed by you except as may be permitted in a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print 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 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 THE END USER 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 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.

Contact CA Technologies

Contact Technical Support

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

- 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

Provide Feedback

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

If you would like 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: Overview 9

Chapter 2: Install the SiteMinder Agent for WebLogic 25

Chapter 6: Configure the SiteMinder Adjudication Provider 91

Chapter 7: Configure Policies 95

Chapter 8: Logging 107

Chapter 9: Verify the SiteMinder Agent Installation and Configuration 115

Chapter 1: Overview

This section contains the following topics:

Introduction

The following sections introduce the SiteMinder Agent for Oracle WebLogic Server and describe how the SiteMinder Agent integrates with this operating environment.

Features of the SiteMinder Agent include:

- SiteMinder integration with the J2EE platform
- Access control of the following J2EE resources:
  - Web Applications (including servlets, HTML pages, JSP, image files)
  - JNDI lookups
  - EJB components
  - JMS connection factories, topics, and queues
  - JDBC connection pools
- Support for SiteMinder single sign-on
- Support for WebLogic clustering

The Agent additionally supports:

- FIPS 140-2
- IPv6
- Centralized and dynamic agent configurations
- Caching of resource protection decisions and authentication and authorization decisions
- Logging

Required Background Information

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

- An understanding of J2EE application servers and multi-tier architecture.
- Familiarity with the WebLogic Security Framework for WebLogic Server.
- Knowledge of how to provide security constraints for J2EE components through deployment descriptors.
- Experience with managing the WebLogic Server, including tasks such as accessing the administrative console.
- Familiarity with SiteMinder concepts, terms, and Policy Server configuration tasks.

SiteMinder Agent Components

The SiteMinder Agent provides a SiteMinder-based access control solution for WebLogic Server by implementing the following security provider modules:

SiteMinder Identity Asserter (IA)

Validates sessions and obtains user credentials from resource requests.

SiteMinder Authentication Provider

Validates user credentials obtained by the SiteMinder Identity Asserter or WebLogic authentication against associated user directories configured in SiteMinder. The SiteMinder Authentication Provider cannot validate credentials obtained from other Identity Asserters.

Determines group membership using the SiteMinders DMS and Identity Manager APIs and populates the subject with a principal for each group returned. Also, returns group membership to the WebLogic Server by using SiteMinder HTTP header responses from the Policy Server.

SiteMinder Authorization Provider

Provides SiteMinder policy-based access decisions based for WebLogic resources.

SiteMinder Adjudication Provider

Provides the final access decision based on decisions made by all authorization providers configured in a WebLogic domain. Resolves any authorization conflicts that occur if authorization providers are configured in addition to the SiteMinder Authorization Provider.

Together, the SiteMinder Agent security provider modules establish a comprehensive trust relationship between the WebLogic Server and SiteMinder.

More Information

SiteMinder Identity Asserter (IA) (see page 11)
SiteMinder Authentication Provider (see page 15)
SiteMinder Authorization Provider (see page 16)
SiteMinder Adjudication Provider (see page 17)

SiteMinder Identity Asserter (IA)

When configured in the WebLogic Security Framework, the SiteMinder Identity Asserter is responsible for handling the following request types:

- HTTP requests from users with preestablished SiteMinder sessions without challenging them for credentials (validating the session and obtaining user names from the associated SiteMinder session ticket cookies).
- HTTP and JAVA requests from users with X.509 certificates (obtaining user names from associated X.509 certificates and authenticating them against SiteMinder user directories).
- HTTP requests from users without preestablished SiteMinder sessions by challenging them for credentials using SiteMinder basic or advanced authentication schemes. A SiteMinder Web Agent provides authentication services for advanced authentication schemes.

Note: The SiteMinder IA always validates requests which contain SiteMinder session cookies or X.509 certificates; configure it to challenge other request types.

The SiteMinder IA then passes a valid user name and SiteMinder session information to an authentication provider for authentication within the WebLogic domain.

Note: If you only need to allow SiteMinder Single Sign-On (SSO) clients to access Web applications, you can use the SiteMinder Identity Asserter as a standalone component without any of the other SiteMinder Agent components.

More information:

Configure the SiteMinder Identity Asserter (see page 57)
Configure the SiteMinder Identity Asserter in SiteMinder (see page 57)
Configure the SiteMinder Identity Asserter in WebLogic (see page 65)

How the SiteMinder Identity Asserter Handles HTTP Requests

The SiteMinder Identity Asserter handles requests for HTTP resources from the following types of users:

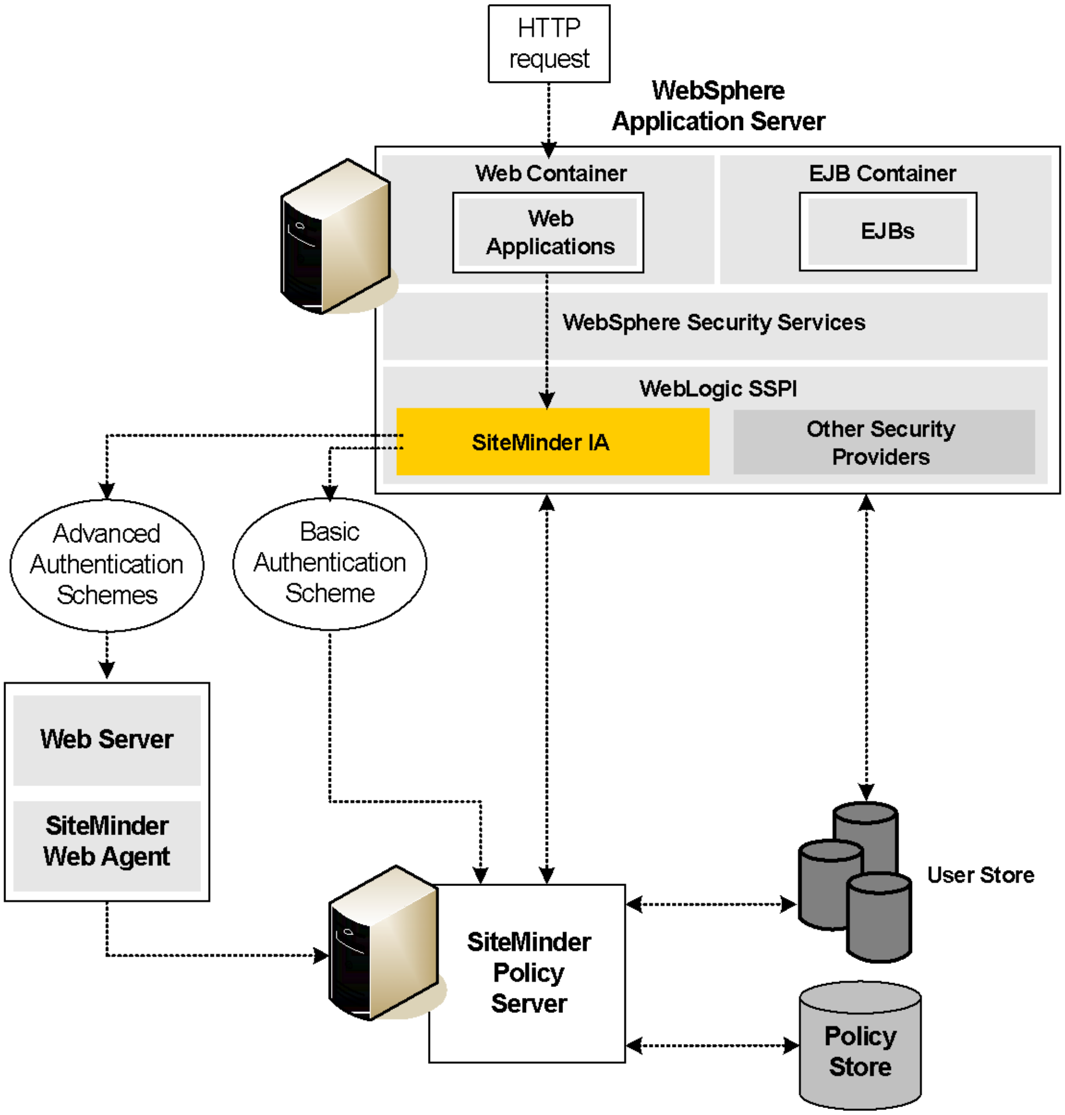
- Users with preestablished SiteMinder sessions (with an SMSESSION cookie) without challenging them for credentials (validating the session and obtaining user names from the associated SiteMinder session ticket cookies).

  Note: SMSESSION cookies can be associated with requests from users with existing SiteMinder Single Sign-On (SSO) sessions or obtained from a Web Agent on a front-end proxy server configured to Intercept HTTP requests, validate user credentials through policies defined on the Policy Server, and forward requests together with user credentials (in a session cookie) to WebLogic.

- Users without preestablished SiteMinder sessions by challenging them for credentials using SiteMinder basic or advanced authentication schemes. A SiteMinder Web Agent provides authentication services for advanced authentication schemes. When configured to challenge requests for credentials, the SiteMinder IA supports the following authentication schemes:

  - Basic
  - Basic over SSL
  - HTML Forms
  - X509 Client Certificate
  - X509 Client Certificate and Basic
  - X509 Client Certificate or Basic
  - X509 Client Certificate and HTML Forms
  - X509 Client Certificate or HTML Forms

Identity Asserter architecture and data flow is shown in the following diagram.

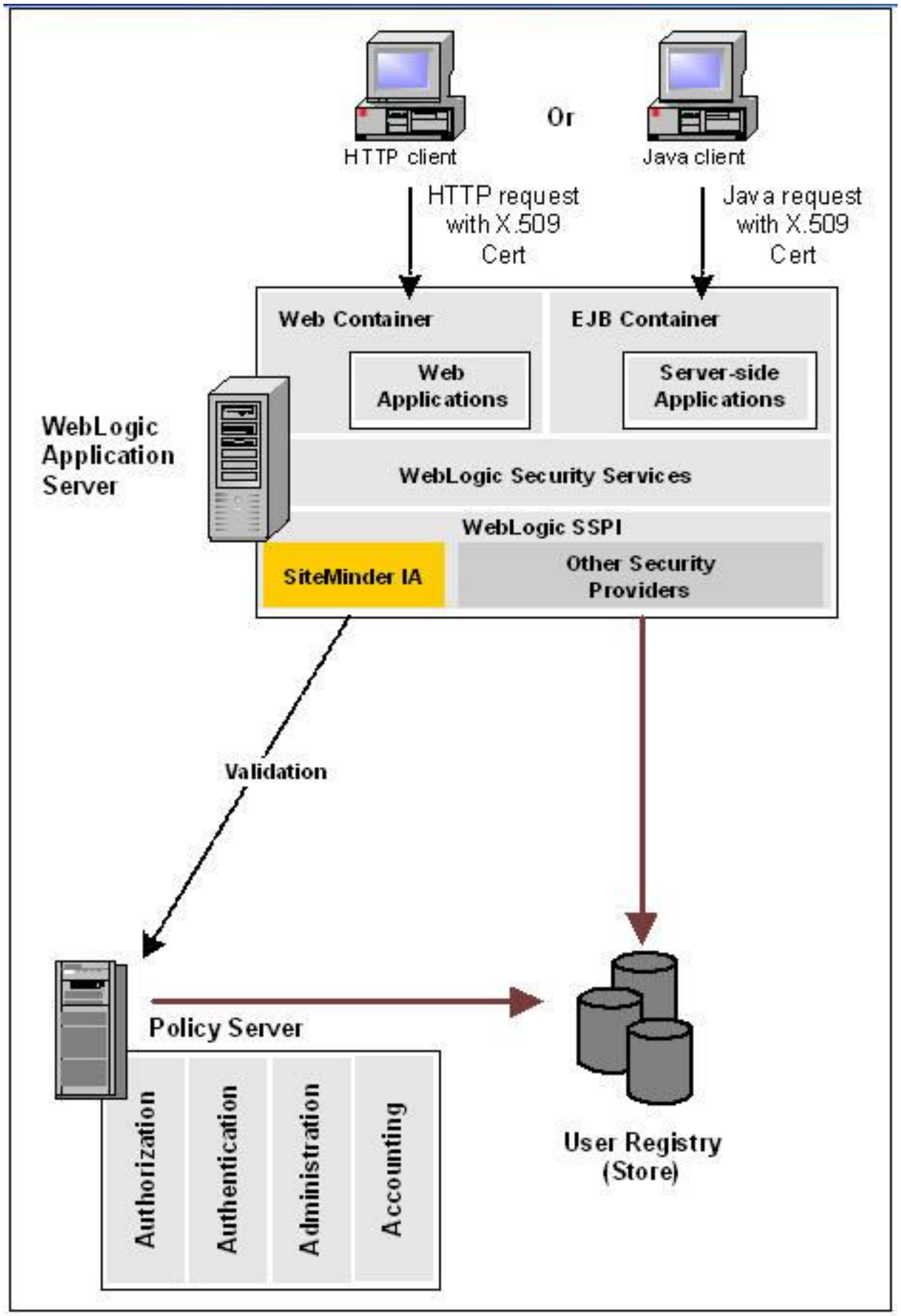


The SiteMinder Identity Asserter always validates requests which contain SiteMinder session cookies; configure it to challenge other requests for credentials.

Note: If you must *only* allow requests with SiteMinder sessions (with an SMSESSION cookie) to access web applications, you can use the SiteMinder Identity Asserter as a standalone component without any of the other SiteMinder Agent components.

How the SiteMinder Identity Asserter Handles Requests with X.509 Certificates

The SiteMinder Identity Asserter handles requests with X.509 certificates obtained from a certificate authority and supplied with any HTTP or Java client request as shown in the following diagram:



SiteMinder Authentication Provider

The SiteMinder Authentication Provider module allows WebLogic to establish trust by validating user credentials against SiteMinder user directories.

The SiteMinder Authentication Provider can handle:

- Requests that originate from the SiteMinder Identity Asserter (which have already been authenticated by SiteMinder and include SiteMinder session information).
- Authentication requests from the WebLogic security layer when the user credentials are collected through HTTP (browser-based) client authentication and Java Client authentication.

The SiteMinder Authentication Provider validates that the username associated with a request maps to a user within the associated user directory configured in SiteMinder.

If SiteMinder authentication is successful, the SiteMinder Authentication Provider populates a WebLogic subject with a SiteMinder principal that contains the username and SiteMinder session data required to prove that SiteMinder authentication has occurred (required by the SiteMinder Authorization Provider).

Groups Module

The SiteMinder Authentication Provider can obtain group information using one or more of the following:

- Identity Manager API (licensed separately)
- DMS API
- SiteMinder responses

You can use a new Agent Configuration Object parameter, SmUserDirectory, which allows the SiteMinder Agent to return group information to the WebLogic Server. If you configure this parameter in the Agent Configuration Object or set the SiteMinder responses so that groups are returned, the SiteMinder Authentication Provider uses one or more of the APIs to obtain the groups that users belong to and populates the subject with a principal for each group.

SiteMinder Authorization Provider

The SiteMinder Authorization Provider determines whether an authenticated user is allowed to access a protected WebLogic resource, based on associated SiteMinder policies configured using the Administrative UI.

Note: The SiteMinder Authorization Provider only accepts subjects populated by the SiteMinder Authentication Provider that contain a principal containing SiteMinder session data (required to prove that SiteMinder authentication has occurred). The SiteMinder Authorization provides an ABSTAIN authorization decision for any other subject passed to it.

Like all WebLogic authorization providers, the SiteMinder Authorization Provider provides PERMIT, DENY, or ABSTAIN authorization decisions based on the policies configured for a particular resource and a number of other contributing factors (as shown in the following table).

In the table, "N/A" denotes either a YES or NO answer that does not affect the final outcome of the authorization decision.

Was Subject Authenticated by SM Auth. Provider?	**Is the Enable-WebAgent parameter set?**	**Exceptions (such as Agent connection problems)?**	**Was the SiteMinder authorization successful?**	**Authorization Decision**
No	No	N/A	N/A	**ABSTAIN**
Yes	No	N/A	N/A	**ABSTAIN**
No	Yes	N/A	N/A	**ABSTAIN**
Yes	Yes	Yes	N/A	**DENY**
Yes	Yes	No	NO	**DENY**
Yes	Yes	No	YES	**PERMIT**

The authorization decision table assumes that the resources in question are protected and that:

- The WebLogic **Abstain if Not Protected** flag is set to "N". With the flag set this way, the SiteMinder Authorization Provider always provides a PERMIT decision for requests for unprotected resources. If the flag is set to "Y", the SiteMinder Authorization Provider always provides an ABSTAIN decision for requests for unprotected resources.
- The WebLogic **Abstain if Not Authenticated** flag is set to "Y". With the flag set this way, the SiteMinder Authorization Provider always provides an ABSTAIN decision for requests not authenticated by SiteMinder. If the flag is set to "N", the SiteMinder Authorization Provider always provides a DENY decision for unauthenticated requests.

Note: If the Authentication Provider is configured, the SiteMinder Adjudication Provider must also be configured.

SiteMinder Adjudication Provider

The SiteMinder Adjudication Provider resolves any authorization conflicts that may occur when more than one authorization provider is configured in a security realm by weighing the result of each authorization provider's access decision. The Adjudication Provider does this by tallying different results returned by multiple Authorization providers' access decisions and providing a final decision on whether access should be granted to a WebLogic resource.

Note: The SiteMinder Adjudication Provider is *required* if the SiteMinder Authorization Provider is configured.

The SiteMinder Adjudication Provider can be configured to operate in two different modes:

- **SiteMinder Precedence**—In this mode, the SiteMinder Adjudication Provider assigns maximum weight to the result returned by a SiteMinder Authorization Provider.
- **Equal Precedence**—In this mode, the SiteMinder Adjudication Provider assigns equal weight to the results returned by all configured Authorization Providers in the security realm.

Note: Do *not* set EnableWebAgent="NO" for the SiteMinder Adjudication Provider—doing so will prevent the WebLogic Server from starting.

The following table indicates the behavior of these modes. In the table, "N/A" denotes either a YES or NO answer that does not affect the final outcome of the authorization decision.

SiteMinder Adjudication Mode	Result from SiteMinder Authorization Provider	Result from other Az Providers configured in the WebLogic Security realm	Authorization Decision
SiteMinder Precedence	ABSTAIN	PERMIT (all)	PERMIT
	ABSTAIN	DENY (one or more)	DENY
	PERMIT	N/A	PERMIT
	DENY	N/A	DENY
Equal Precedence	ABSTAIN	PERMIT (all)	PERMIT
	ABSTAIN	DENY (one or more)	DENY
	DENY	PERMIT (one or more)	DENY
	PERMIT	DENY (one or more)	DENY
	PERMIT	PERMIT (all)	PERMIT

Which SiteMinder Security Providers Do I Need?

The SiteMinder security provider modules you require depend on your WebLogic access control needs. Select the security provider modules according to the functionality you require, being careful to verify that the upstream and downstream requirements (that is, requirements from elements before and after in the flow of data in the security framework) of security providers match up as shown in the following table.

Security Provider	Upstream Requirements	Downstream Requirements
SiteMinder Identity Asserter (for SMSESSION cookies)	A trusted issuer of SiteMinder session cookies.	None.

Security Provider	**Upstream Requirements**	**Downstream Requirements**
SiteMinder Identity Asserter (for X.509 certificates)	A trusted issuer of X.509 certificate tokens.	Requires SiteMinder Authentication Provider to authenticate identities obtained from X.509 certificates.
SiteMinder Identity Asserter (for challenged requests)	None.	SiteMinder Authentication Provider to authenticate credentials obtained by the configured authentication scheme.
SiteMinder Authentication Provider	Requires SiteMinder Identity Asserter to validate and obtain user identity and SiteMinder session information from SiteMinder session cookies and X.509 certificates. Does **not** accept users obtained from other Identity Asserters.	None.
SiteMinder Authorization Provider	Requires subject populated by SiteMinder Authentication provider (containing a SiteMinder principal). ABSTAINs from other authorization decisions.	Requires SiteMinder Adjudication Provider to resolve authorization disputes with other authorization providers.
SiteMinder Adjudication Provider	Requires SiteMinder Authorization Provider to be one of the configured authorization providers.	-N/A-

However, it is likely that most deployments fall into one of the following two scenarios:

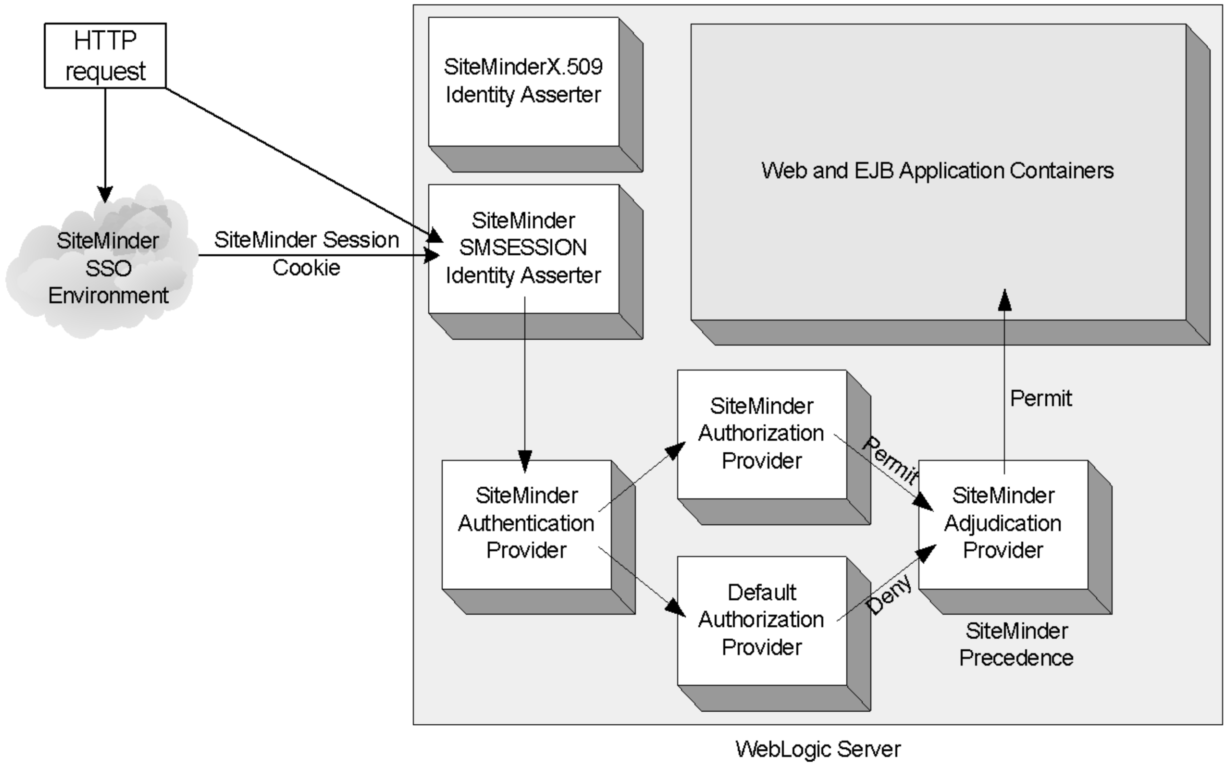
Problem	Solution
You need to establish a trust relationship between the SiteMinder and WebLogic Single-Sign On (SSO) environments so that HTTP clients authenticated by SiteMinder are not rechallenged by WebLogic when they access Web applications hosted by a WebLogic Server. You have existing WebLogic or application-based authorization policies that are sufficient for your needs.	Configure just the SiteMinder Identity Asserter.
You need to implement SiteMinder authentication and authorization policies for all requests for Web and server-side applications.	Configure the complete SiteMinder Agent solution, comprising of: ■ SiteMinder Authentication Provider ■ SiteMinder Authorization Provider ■ SiteMinder Adjudication Provider ■ SiteMinder Identity Asserter (optional, if perimeter authentication required)

Use Cases

The following use cases illustrate the use of different SiteMinder Agent components to solve different access control needs.

HTTP Requests Using All SiteMinder Security Providers Use Case

In the configuration illustrated in the following diagram, the SiteMinder SMSESSION Identity Asserter handles requests for Web container applications (with or without associated SiteMinder session cookies if configured to challenge for credentials).


HTTP request
SiteMinder SSO Environment
SiteMinder Session Cookie
SiteMinderX.509 Identity Asserter
SiteMinder SMSESSION Identity Asserter
Web and EJB Application Containers
SiteMinder Authentication Provider
SiteMinder Authorization Provider
Default Authorization Provider
Permit
Deny
Permit
SiteMinder Adjudication Provider
SiteMinder Precedence
WebLogic Server

The SiteMinder Authentication Provider validates user credentials obtained by the SiteMinder Identity Asserter against associated user directories configured in SiteMinder.

The SiteMinder Authorization Provider provides SiteMinder authorization for all requests.

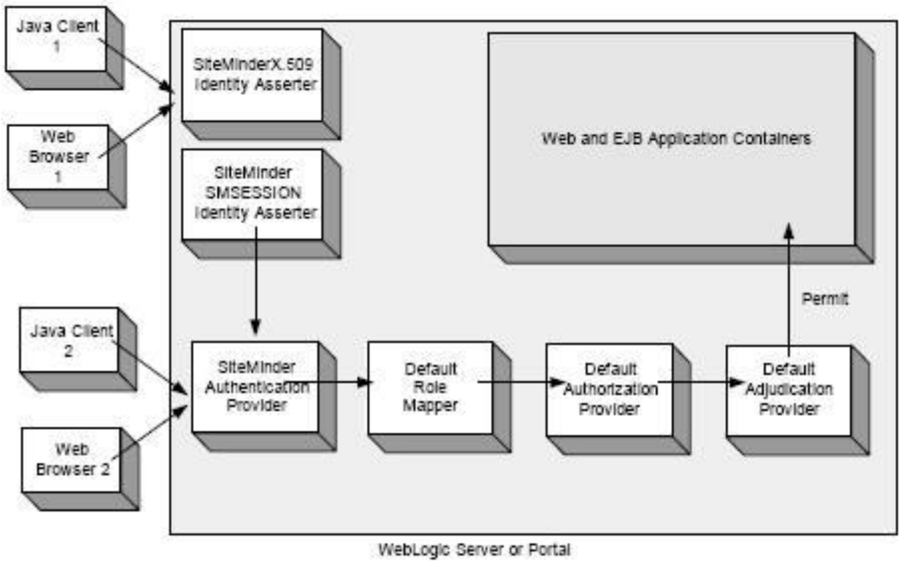
The SiteMinder Adjudication Provider Provides the final access decision and resolves any authorization conflicts that occur if authorization providers are configured in addition to the SiteMinder Authorization Provider.

SiteMinder Authentication Provider Use Cases

In the following figure, the SiteMinder Authentication Provider interacts with the WebLogic Server default role mapper and authorization provider. The SiteMinder Authentication Provider returns groups that are mapped to roles by the default role mapper. These roles are then used by the default authorization provider for authorization decisions.

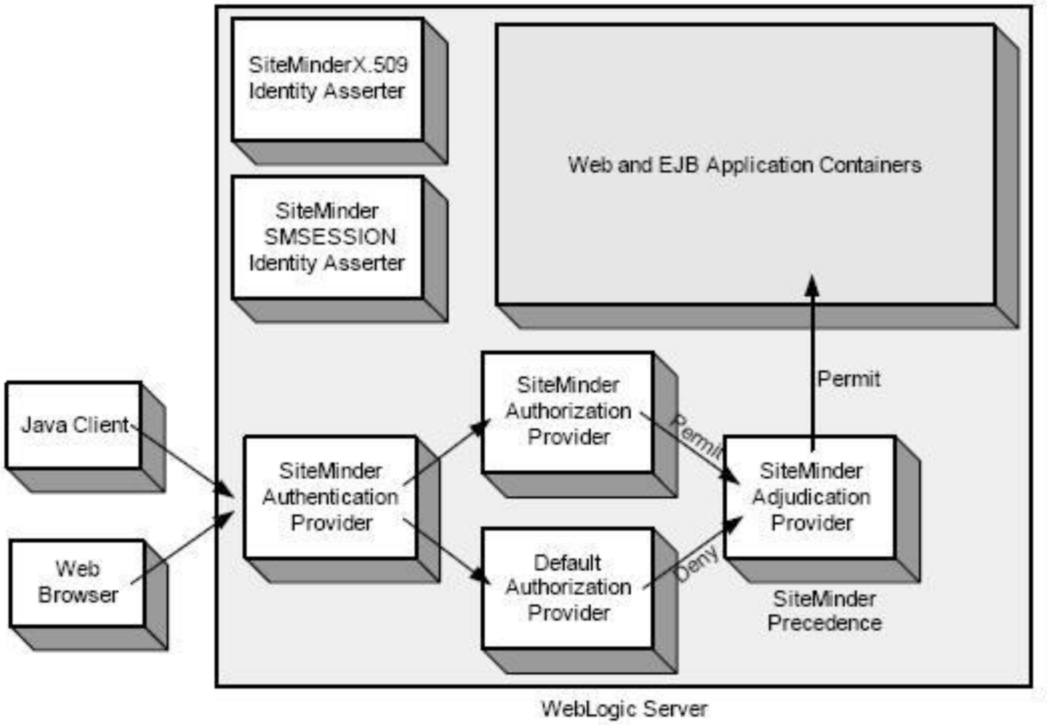
In one use case, Java Client 1 and Web Browser 1 go through an Identity Asserter and make requests directly to the container. The requests have an identity assertion token.

In a second use case, no Identity Asserter is available and Java Client 2 and Web Browser 2 make requests directly to the container. The requests do not contain an identity assertion token and users must enter a password.


Java Client 1
Web Browser 1
Java Client 2
Web Browser 2
SiteMinderX.509 Identity Asserter
SiteMinder SMSESSION Identity Asserter
SiteMinder Authentication Provider
Default Role Mapper
Default Authorization Provider
Default Adjudication Provider
Web and EJB Application Containers
Permit
WebLogic Server or Portal

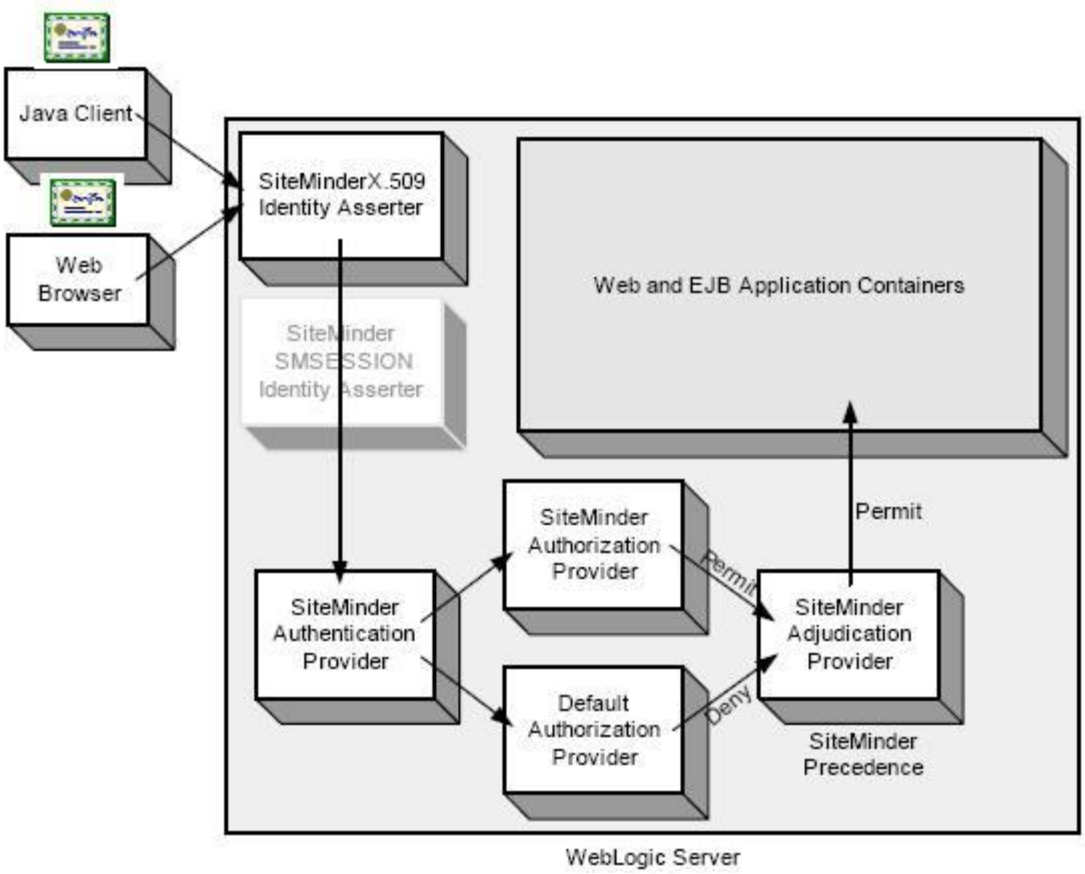
No Identity Asserter Use Case

In the configuration illustrated in the following figure, requests from Web and Java clients are made directly to the container. WebLogic collects credentials which are then handled by the SiteMinder Authentication Provider.


SiteMinderX.509 Identity Asserter
SiteMinder SMSESSION Identity Asserter
Web and EJB Application Containers
Permit
Java Client
Web Browser
SiteMinder Authentication Provider
SiteMinder Authorization Provider
Permit
Default Authorization Provider
Deny
SiteMinder Adjudication Provider
SiteMinder Precedence
WebLogic Server

X.509 Identity Asserter Use Case

In the configuration illustrated in the following figure, the SiteMinder X.509 Identity Asserter obtains credentials from the certificates associated with Web or Java client request and passes those on to the SiteMinder Authentication Provider for authentication.


Java Client
Web Browser
SiteMinderX.509 Identity Asserter
SiteMinder SMSESSION Identity Asserter
Web and EJB Application Containers
SiteMinder Authentication Provider
SiteMinder Authorization Provider
Default Authorization Provider
Permit
Deny
Permit
SiteMinder Adjudication Provider
SiteMinder Precedence
WebLogic Server

Chapter 2: Install the SiteMinder Agent for WebLogic

This section contains the following topics:

Introduction

The following sections describe how to install the SiteMinder Agent for WebLogic Server on Windows and UNIX platforms. The SiteMinder Agent installation includes the following security providers:

- Identity Asserter (IA)
- Authentication Provider
- Authorization Provider
- Adjudication Provider

Although each of these providers is installed when you run the SiteMinder Agent installation, you need only configure the providers that you want to use.

More Information

Configure the SiteMinder Identity Asserter (see page 57)
Configure the SiteMinder Authentication Provider (see page 71)
Configure the SiteMinder Authorization Provider (see page 85)
Configure the SiteMinder Adjudication Provider (see page 91)

ASA_HOME

Throughout this guide, *ASA_HOME* refers to the installed location of the SiteMinder Agent. For example:

Windows: *ASA_HOME*=c:\smwlsasa

UNIX: *ASA_HOME*=/opt/smwlsasa

WLS_HOME

Throughout this guide, WLS_*HOME* refers to the installed location of the WebLogic Server.

Upgrade from a Previous Release

The SiteMinder Agent for Oracle WebLogic software cannot be upgraded from a previous version. To install the current version, first uninstall any previous version of the SiteMinder Agent (or Application Server Agent) for WebLogic. For more information, see the Agent Guide associated with the release that you must uninstall.

Software Requirements

Install supported versions of required software before you install the SiteMinder Agent.

For a complete list of supported software, operating systems, Java environments, and prerequisite CA product versions, see the SiteMinder Agent for Application Servers Platform Support Matrix on the Technical Support site.

Requirements for all installations

Supported versions of the following software must be installed and properly configured before you install the SiteMinder Agent:

- 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) is installed and the path to the JRE is present in your environment. For example, on UNIX systems, if your JRE is not in the PATH variable, run these commands:

  PATH=$PATH:*JRE***/bin**

  export PATH

  JRE

  Specifies the location of the Java Runtime Environment.

  Note: For a list of supported JREs, see the SiteMinder support matrix on the Technical Support site.
- SiteMinder Policy Server (typically on a different system in production environments).

Requirements for installations featuring advanced authentication

To use the SiteMinder Identity Asserter to challenge web requests that do not include a valid SiteMinder session cookie for credentials using advanced (other than Basic) authentication schemes, a supported SiteMinder Web Agent must also be installed.

Note: The SiteMinder Policy Server and Web Agent (where applicable) can be installed on different systems than the WebLogic Server.

More Information

Installation Checklist (see page 28)
Install a Web Agent for Advanced Identity Asserter Authentication (see page 41)

Installation Checklist

Before you install the SiteMinder Agent on the WebLogic Server, complete the steps in the following table. To help 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...
	1. Install and configure the SiteMinder Policy Server.	*SiteMinder Policy Server Installation Guide*
	2. Install the WebLogic Server.	The Oracle WebLogic Server documentation.
	3. Configure the Policy Server for the SiteMinder Agent.	Configure the SiteMinder Policy Server for the SiteMinder Agent Providers
	4. Install the SiteMinder Agent on WebLogic Server For WebLogic clusters, install the SiteMinder Agent on each node in the cluster.	Install the SiteMinder Agent (see page 30)
	5. If using the SiteMinder Identity Asserter to challenge requests for credentials using advanced authentication schemes, install and configure a SiteMinder Web Agent.	Install a Web Agent for Advanced Identity Asserter Authentication (see page 41)
	6. Perform additional required configuration steps.	Post Installation Steps

Configure the SiteMinder Policy Server for the SiteMinder Agent Providers

Before you install the SiteMinder Agent, configure SiteMinder objects for the SiteMinder Agent in the Administrative UI.

The Agent objects used by SiteMinder Agent for WebLogic fully conform to the SiteMinder central Agent management model. Configure the following agent objects in a manner similar as you would for a SiteMinder Web Agent:

- Host Configuration Object (typically one for each application server host)
- Agent Configuration Object (one for the SiteMinder Agent providers)
- Agent identity (one for the SiteMinder Agent providers)

Note: For detailed information about how to configure Agent-related objects (Web Agent and other SiteMinder Agents), see *CA SiteMinder Policy Server Configuration Guide* and the *CA SiteMinder Web Agent Installation Guide*.

To configure the SiteMinder Policy Server for the SiteMinder Agent

On the Policy Server:

1. 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.

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

3. Create an Agent identity for the Agent. Select **Web Agent** as the Agent type for an SiteMinder Agent.

4. Duplicate or create an Agent Configuration Object for the SiteMinder Agent. The Agent Configuration Object holds Agent configuration parameters and can be used to configure a group of Agents centrally.

   Note: The SiteMinder Agent for Oracle WebLogic does not use the same agent configuration parameters as a SiteMinder Web Agent and even where parameters have similar names their values might not be compatible. Do not attempt to use the Agent Configuration Object for a SiteMinder Web Agent for the SiteMinder Agent for WebLogic. For a complete listing of configuration parameters for the SiteMinder Agent, see Agent Configuration Parameters (see page 125).

5. Add or edit Agent parameters in that Agent Configuration Object.

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

   In addition, the Agent accepts several optional configuration parameters.

After configuring the Policy Server for the SiteMinder Agent, install the SiteMinder Agent software. Then, complete the post installation steps.

More Information

SiteMinder Agent Installation and Configuration Files (see page 123)
Install the SiteMinder Agent (see page 30)
SiteMinder Agent Directory Structure (see page 123)

Install the SiteMinder Agent

The following sections contain information about installing the SiteMinder Agent.

Installation Requirements

Before you install the SiteMinder Agent:

- Uninstall any previously installed SiteMinder Agent versions.
- Configure the Policy Server for the SiteMinder Agent Providers.

  Verify that a supported Java Runtime Environment (JRE) is installed and the path to the JRE is present in your environment.

- Patch the JRE used by the SiteMinder Agent and WebLogic Server for unlimited cryptography with the Java Cryptography Extension (JCE) package.

  Note: If the JRE is not patched to support unlimited key strength, host registration will fail during SiteMinder Agent installation and the SiteMinder Agent will not work.

- Verify that you have the following information, as you will be prompted for it during the installation:
  - Installation location of the WebLogic Application Server
  - Policy Server IP Address
  - Information about the Trusted Host:

    To register a new Trusted Host, you need the name of the Trusted Host Configuration Object that you created when you configured the SiteMinder Policy Server for the SiteMinder agent providers.

    Note: If you want to register a new Trusted Host, be sure that the Policy Server is running before you start the SiteMinder Agent installation.

    To use an existing Trusted Host on the physical computer where the SiteMinder Agent resides, you need the location of the SmHost.conf file.

    Note: Only use an existing Trusted Host if you are reinstalling the SiteMinder Agent for WebLogic and the SmHost.conf file that you want to use was therefore created by the smreghost tool supplied with this release. The SiteMinder Agent for WebLogic is implemented using a pure Java SiteMinder Agent API and cannot use an SmHost.conf file created for another SiteMinder Agent to establish its connection to the Policy Server.
  - Agent Configuration Object name for the Agent you created when you configured the SiteMinder Policy Server for the SiteMinder agent providers

More Information

Software Requirements (see page 27)

Define the JAVA_HOME Environment Variable

The SiteMinder Agent install and uninstall programs require that a JAVA_HOME variable is defined in the environment that specifies the installed location of the Java Runtime Environment (JRE).

To set the JAVA_HOME variable on Windows

1. Open a command window.
2. Enter the following command:

```
set JAVA_HOME=JRE
```

JRE

Defines the location of the Java Runtime Environment install directory.

To set the JAVA_HOME variable on UNIX

1. Open a command shell.
2. Run the following command:

```
set JAVA_HOME=JRE ; export JAVA_HOME
```

JRE

Defines the location of the WebSphere Java Runtime Environment install directory.

Upgrade the SiteMinder Agent

You cannot upgrade from any other previous versions of the SiteMinder Agent for WebLogic. Uninstall any other previous versions and then install this version of the SiteMinder Agent on your WebLogic Server.

Also, do not attempt to install this version of the SiteMinder Agent on a version of WebLogic other than the one on which it is specifically supported.

Installation Options

This section describes the options for installing the SiteMinder Agent.

Windows:

Run the installation in the graphical user interface (GUI) mode to install the SiteMinder Agent.

UNIX:

Do one of the following to install or upgrade the SiteMinder Agent:

- Use the graphical user interface (GUI) mode.
- Use the console mode.

More Information

Uninstall the Agent (see page 53)

Run the Installation in GUI Mode

The SiteMinder Agent installation program installs all the necessary files for running the SiteMinder Agent. You can run the installation program for the SiteMinder Agent for Oracle WebLogic using a graphical user interface on Windows and UNIX platforms.

The installation program and other required files can be downloaded from the Technical Support site.

To obtain the installation kit from the Support site

1. Click Technical Support.
2. Log in to CA Support Online.
3. Click Download Center.
4. Search the Download Center for the CA SiteMinder Agent for WebLogic installation kit for your operating environment.
5. Download the kit and extract its content to a temporary location.
6. Verify that all required files are present:
   - ca-asa-was-12.0-sp02.bat (Windows)
   - ca-asa-was-12.0-sp02.sh (UNIX)
   - ca-asa-was-install.jar

Notes for UNIX Installations

If you are planning to run the installation in GUI mode on UNIX, consider the following before you begin:

- Running a GUI-mode installation or running the Configuration Wizard using the Exceed application can cause truncation of text in dialogs because of unavailable fonts. This limitation has no affect on SiteMinder Agent installation and configuration.
- If you are installing the SiteMinder Agent over telnet or other terminal emulation software, you must have an X-Windows session running in the background to run the GUI mode installation. Additionally, set the DISPLAY variable to your terminal, as follows:

  ```
  DISPLAY=111.11.1.12:0.0
  ```

  ```
  export DISPLAY
  ```

  If you try to run in GUI mode through a telnet window without an X-Windows session, the installer throws a Java exception and exits.
- You can also run a command line installation from a console window.

To install the SiteMinder Agent using the graphical user interface (GUI) mode

1. Login as the user who installed WebLogic. For example, if you installed as root, login as root.
2. Exit all applications that are running.
3. Open a command window and navigate to where the install kit is located
4. Enter the appropriate command for your operating system.

   Windows:

   ```
   ca-asa-wls-12.0-sp02.bat
   ```

   UNIX:

   ```
   sh ca-asa-wls-12.0-sp02.sh
   ```
5. Read the License Agreement. If you accept the terms, select the I accept the terms of the License Agreement option and click Next.

6. On the Choose Install Folder panel, specify a location for installing the SiteMinder Agent for Oracle WebLogic and click Next. We recommend the following default location:

   Windows: *drive*:\smwlsasa

   UNIX: /opt/smwlsasa

   If you specify a folder that does not exist, the installer asks if you want to create it. Click Yes to create it; the installer creates a folder named smwlsasa in whatever directory you specify.

   The program installs the required files.

   On the Choose WebLogic Folder screen, specify the installation location of the WebLogic Server and click Install. The program installs the required files.

   Note: If the location you specify is not present, the installation program displays an error message and asks you to reenter the information.

7. In the Host Registration dialog, select one of the following:

   - Yes, create trusted host — The installer invokes the Host Registration tool, smreghost, to register the unique trusted host name with the Policy Server and create the SmHost.conf file. Registering the system as a trusted host enables the SiteMinder Agent to establish a secure, trusted connection with the Policy Server.

     Create a Host Configuration Object in the Administrative UI before registering a trusted host.

   - No, use existing file — The installer invokes the smreghost tool to use an existing SmHost.conf file created for a SiteMinder Agent for WebLogic to establish the connection between the trusted host and the Policy Server.

     Note: Specify this option *only* if you are reinstalling the SiteMinder Agent for WebLogic and the SmHost.conf file that you want to use was therefore created by the smreghost tool supplied with this release. The SiteMinder Agent for WebLogic is implemented using a pure Java SiteMinder Agent API and cannot use an SmHost.conf file created for another SiteMinder Agent to establish its connection to the Policy Server.

8. If you selected "Yes, create a trusted host" on the Host Registration screen, do the following:

   a. On the FIPS Mode Setting screen, click one of the following options and then click Next:

      FIPS Compatibility Mode (Default)

      Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing 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. If you do *not* want to use FIPS encryption, accept this default.

      FIPS Migration Mode

      Specifies FIPS-migration mode, which is used when you are upgrading an earlier version of SiteMinder to full-FIPS mode. The Policy Server and the Agents continue to use the existing SiteMinder encryption algorithms as you migrate your environment to use only FIPS 140-2 approved algorithms.

      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 SiteMinder installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of SiteMinder that do not support FIPS, 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. Relink all custom software using Policy Management APIs or any other API that the Policy Server exposes with FIPS-supporting versions of the respective SDKs to achieve the required support for Full FIPS mode.

   b. On the the Host Registration screen, enter the following information and click Next:

      - Policy Server IP Address—IP address of the Policy Server where you are registering the host
      - SM Admin Username—Name of the administrator permitted to register the host with the Policy Server
      - SM Admin Password—Password for the SM Admin account

- Host Name—Unique name that represents the trusted host to the Policy Server. The name does not have to be the same as the physical client system you are registering; it can be any unique name.
- Host Config Object— Name of the Host Configuration Object specified in the Policy Server.

The installation program registers your unique trusted host name with the Policy Server. If your Policy Server is not running, an error message appears and you can register the trusted host later using the smreghost tool.

If you have not patched the JVM Java Cryptography Extension (JCE) package for unlimited cryptography, host registration fails and the following error message appears:

```
Failed to enable any clusters. Registration has failed.
```

9. If you selected "No, use existing file" on the Host Registration screen, enter the location of a host configuration file (SmHost.conf) created for a SiteMinder Agent for WebSphere in the text box, or click Choose to browse for the file.

   The default location of SmHost.conf is either:

   ASA_HOME\conf\ (Windows)

   or

   ASA_HOME/conf/ (UNIX)

10. On the Agent Configuration screen, specify the name of the Agent Configuration Object that you created in the Administrative UI before installing the SiteMinder Agent. Click Next.
11. In the Install Complete dialog, click Done to exit the installer.

    The installation is complete.

12. Restart the WebLogic Server for installation changes to take effect.

More Information

SiteMinder Agent Directory Structure (see page 123)

Run the Installation in Console Mode on UNIX

When performing a fresh install on UNIX platforms, you can run the installation program for the SiteMinder Agent for Oracle WebLogic from the console.

The installation program and other required files can be downloaded from the Technical Support site.

To obtain the installation kit from the Support site

1. Click Technical Support.
2. Log in to CA Support Online.
3. Click Download Center.
4. Search the Download Center for the CA SiteMinder Agent for WebLogic installation kit for your operating environment.
5. Download the kit and extract its content to a temporary location.
6. Verify that all required files are present:
   - ca-asa-was-12.0-sp02.bat (Windows)
   - ca-asa-was-12.0-sp02.sh (UNIX)
   - ca-asa-was-install.jar

To install the SiteMinder Agent for WebLogic by running the installation script in a UNIX console

1. Login as the user who installed WebLogic. For example, if you installed as root, login as root.
2. Exit all applications that are running.
3. Open a command shell and navigate to where the install kit is located
4. Enter the following command:

   ```
   sh ca-asa-wls-12.0-sp02.sh -i console
   ```

   The -i console option interactively runs the installation from a console.
5. Read the License Agreement. If you accept the terms, enter Y and then press Enter.
6. In the Choose Install Folder section, specify a location for the SiteMinder Agent for Oracle WebLogic installation, and then press Enter.

   We recommend the following location:

   /opt/smwlsasa

7. Enter **Y**, then press Enter to create or confirm the installation location for the SiteMinder Agent.

   The program installs the required files in the SiteMinder Agent install location.

   The program installs the required files in the WebLogic install location.

8. Specify the installation location of the WebLogic Server.

   The program installs the required files in the WebLogic install location.

9. When the Host Registration prompt appears, select one of the following numbers:

   - **1**—The installer invokes the Host Registration tool, smreghost, to register the unique trusted host name with the Policy Server and create the SmHost.conf file. Registering the system as a trusted host enables the SiteMinder Agent to establish a secure, trusted connection with the Policy Server.

     Create a Host Configuration Object in the Administrative UI before registering a trusted host.

   - **2**—The installer invokes the smreghost tool to use an existing SmHost.conf file created for a SiteMinder Agent for WebLogic to establish the connection between the trusted host and the Policy Server.

     Note: Specify this option *only* if you are the reinstalling the SiteMinder Agent for WebLogic and the SmHost.conf file that you want to use was therefore created by the smreghost tool supplied with this release. The SiteMinder Agent for WebLogic is implemented using a pure Java SiteMinder Agent API and cannot use an SmHost.conf file created for another SiteMinder Agent to establish its connection to the Policy Server.

10. If you entered 1 at the Host Registration prompt (to create a new trusted host), do the following:

    a. When prompted to select a FIPS mode, choose one of the following options:

       - **1**—FIPS Compatibility Mode (Default)

         Specifies non-FIPS mode, which lets the Policy Server and the Agents read and write information using the existing 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. If you do *not* want to use FIPS encryption, accept this default.

- **2**—FIPS Migration Mode

  Specifies FIPS-migration mode, which is used when you are upgrading an earlier version of SiteMinder to full-FIPS mode. The Policy Server and the Agents continue to use the existing SiteMinder encryption algorithms as you migrate your environment to use only FIPS 140-2 approved algorithms.

- **3**—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 SiteMinder installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of SiteMinder that do not support FIPS, 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. Relink all custom software using Policy Management APIs or any other API that the Policy Server exposes with FIPS-supporting versions of the respective SDKs to achieve the required support for Full FIPS mode.

b. When prompted, enter the following information:

   - Policy Server IP Address–IP address of the Policy Server where you are registering the host
   - SM Admin Username–Name of the administrator permitted to register the host with the Policy Server
   - SM Admin Password–Password for the SM Admin account
   - Host Name–Unique name that represents the trusted host to the Policy Server. The name does not have to be the same as the physical client system you are registering; it can be any unique name.
   - Host Config Object–Name of the Host Configuration Object specified in the Policy Server.

   The installation program registers your unique trusted host name with the Policy Server. If your Policy Server is not running, a message appears and you can register the trusted host manually later.

   If you have not patched the JVM Java Cryptography Extension (JCE) package for unlimited cryptography, host registration fails and the following error message appears:

   ```
   Failed to enable any clusters. Registration has failed.
   ```

11. If you entered "2" to use an existing trusted host, enter the location of the host configuration file (smhost.conf) created for a SiteMinder Agent for WebLogic.

    The default location of the file is:

    ASA_HOME/conf/

12. Supply the name of the Agent Configuration Object that you created for the SiteMinder Agent.
13. At the installation complete prompt, press Enter to exit the installer. The installation of the SiteMinder Agent for Oracle WebLogic is complete.
14. Restart the WebLogic Server for installation changes to take effect.

More Information

SiteMinder Agent Directory Structure (see page 123)

Install a Web Agent for Advanced Identity Asserter Authentication

If you are configuring the SiteMinder Identity Asserter to challenge requests for credentials, a SiteMinder Web Agent is required to collect credentials and authenticate user requests for authentication schemes other than Basic (the Identity Asserter can handle basic authentication itself).

If no suitable Web Agent is present in your SiteMinder environment, install and configure one (together with a supported Web Server).

For information about how to install and configure SiteMinder Web Agents, see the *CA SiteMinder Web Agent Installation Guide* and the *CA SiteMinder Agent Guide*.

More information:

Installation Checklist (see page 28)
SiteMinder Identity Asserter (IA) (see page 11)

Register a Trusted Host Using the Registration Tool

When you install a SiteMinder Agent on a server for the first time, you are prompted to register that server as a trusted host. Once the trusted host is registered, you do not have to reregister with subsequent Agent installations.

There might be situations when you want to register or reregister a trusted host independent of an Agent installation, such as the following:

- You chose not to register the trusted host during Agent installation
- You must change the FIPS mode the SiteMinder Agent and Policy Server use to exchange information
- To rename the trusted host if there has been a change to your SiteMinder environment
- To reestablish a trusted host if the trusted host has been deleted in the Administrative UI
- To recreate policy objects 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

Register a Trusted Host on Windows

To register or reregister a trusted host on Windows, use the Registration Tool, smreghost. This tool is installed when you install an Agent on a trusted host, and is located in the directory *ASA_HOME*\bin.

Note: When reregistering a host with the same name using smreghost, first remove the host from the Administrative UI unless you use the smreghost command argument, **-o**, which lets you overwrite an existing trusted host without having to delete it from the Policy Server.

To run smreghost to register or reregister a trusted host on Windows

1. Open a Command Prompt window.
2. Navigate to *ASA_HOME*\bin
3. Enter the smreghost command using at least 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
```

   The smreghost also takes a number of optional requirements not shown here. For a complete list of smreghost arguments, see smreghost Command Arguments (see page 44).

   Note: There must be a space between each command argument and its value.

More information:

smreghost Command Arguments (see page 44)
smreghost Command Examples (see page 46)

Register a Trusted Host on UNIX

To register or reregister a trusted host on UNIX use the Registration Tool, smreghost. This tool is installed when you install an Agent on a trusted host, and is located in the directory *ASA_HOME*/bin.

Note: When reregistering a host with the same name using smreghost, first remove the host from the Administrative UI unless you use the smreghost command argument, **-o**, which lets you overwrite an existing trusted host without having to delete it from the Policy Server.

To run smreghost to register or reregister a host on UNIX

1. Open a Command Prompt window.
2. Verify that the library path environment variable contains the path to the SiteMinder Agent bin directory by entering the following two commands:

   ```
   library_path_variable=${library_path_variable}:ASA_HOME/bin

   export library_path_variable
   ```

   where *library_path_variable* is LD_LIBRARY_PATH for Solaris and Linux and is SHLIB_PATH for HP-UX.

   Example: setting the library path

   To set the library path for Solaris systems, enter the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/smwlsasa/bin

   export LD_LIBRARY_PATH
   ```

3. Enter the smreghost command using at least 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
   ```

   The smreghost also takes a number of optional requirements not shown here. For a complete list of smreghost arguments, see smreghost Command Arguments (see page 44).

   Note: There must be a space between each command argument and its value.

More information:

smreghost Command Arguments (see page 44)
smreghost Command Examples (see page 46)

smreghost Command Arguments

This following is a complete list of valid arguments for the smreghost tool.

-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 servers (authentication, authorization, accounting), however, the unified server responds to any Agent request on any port. For example, if you specify port 55555, the policy server entry in the SmHost.conf file will show the following:

"*policy_server_ip_address*,5555,5555,5555"

Example: (IPv4) 127.0.0.1,55555

Example: (IPv6) [2001:DB8::/32][:55555]

-u *administrator_username*

Indicates Name of the 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 Web 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 backups 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 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.

- MIGRATE--Specifies FIPS-migration mode, which is used when you are upgrading an earlier version of SiteMinder to full-FIPS mode. The Policy Server and the Agents continue to use the existing SiteMinder encryption algorithms as you migrate your environment to use only FIPS 140-2 approved algorithms.

- 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 SiteMinder installation that is running in Full FIPS mode cannot interoperate with, or be backward compatible to, earlier versions of 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 SiteMinder Cryptographic Boundary exists in the *Policy Server Administration Guide*.

Note: Stop the WebSphere profile before registering the SiteMinder Agent in FIPS-migratation mode.

-cp *cryptographic_provider*

(Optional) Indicates the name of the cryptographic provider you are using for encryption. If you do not specify a value the default is assumed.

Default: ETPKI

-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.

smreghost Command Examples

The following examples show valid smreghost commands for different host registration scenarios.

Register a Host Example

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn hostA
    -hc DefaultHostSettings
```

Register a Host in FIPS-only Mode Example

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn hostA
    -hc DefaultHostSettings -cf ONLY
```

Reregister a Host Example

```
smreghost -i 123.123.1.1 -u SiteMinder -p mypw -hn hostA

    -hc DefaultHostSettings -o
```

Post Installation Steps

Once you have installed the SiteMinder Agent, complete these steps:

1. Set up home and classpath environment variables.
2. Set up the Agent configuration file. (WebAgent.conf).
3. Configure the following SiteMinder Agent providers as needed.
4. Set up policies for the SiteMinder Agent with the SiteMinder Policy Server.

Note: For information about creating policies, see the *SiteMinder Policy Server Configuration Guide.*

More Information

Set the WebLogic Environment for the SiteMinder Agent (see page 47)
Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Configure the SiteMinder Identity Asserter (see page 57)
Configure the SiteMinder Adjudication Provider (see page 119)
Configure the SiteMinder Authentication Provider (see page 71)
Configure the SiteMinder Authorization Provider (see page 85)
Configure Policies (see page 95)

Set the WebLogic Environment for the SiteMinder Agent

Before the SiteMinder Agent can operate with the WebLogic Server, you must configure SiteMinder Agent-related environment settings in one of the following:

- 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 documentation.

More information:

ASA_HOME (see page 26)

Set the WebLogic Environment for SiteMinder on Windows

To set the environment for the SiteMinder Agent on a standalone WebLogic Server on Windows, do the following:

- Define a Java environment variable smasa.home that refers to the directory where the SiteMinder Agent for WebLogic is installed.
- Add the following SiteMinder Agent files and directories to the CLASSPATH variable:
  - *ASA_HOME*\conf
  - *ASA_HOME*\lib\smagentapi.jar
  - *ASA_HOME*\lib\smjavasdk2.jar
  - *ASA_HOME*\lib\sm_cryptoj.jar
  - *ASA_HOME*\lib\smclientclasses.jar
- Add the following SiteMinder Agent option to the WebLogic execution entry:
  - -Dsmasa.home=%SMASA_HOME%

Example: Set the Environment for a Standalone WebLogic Server on Windows

The following procedure is an example of how to set the environment for the SiteMinder Agent on a standalone WebLogic Server on Windows.

1. Edit the startWebLogic.cmd file.

   The startWebLogic.cmd is located in *wl_install*\user_projects\domains\your_domain\bin

   where *wl_install* is the installed location of the WebLogic application server, and *your_domain* is the name of the WebLogic domain where the SiteMinder Agent is installed.

2. Define SMASA_HOME as follows:

   ```
   set SMASA_HOME=ASA_HOME
   ```

   ASA_HOME

   Specifies the installed location of the SiteMinder Agent.

3. Define SMASA_CLASSPATH as follows:

   ```
   set SMASA_CLASSPATH=%SMASA_HOME%\conf;
   %SMASA_HOME%\lib\smagentapi.jar;
   %SMASA_HOME%\lib\smjavasdk2.jar;
   %SMASA_HOME%\lib\sm_cryptoj.jar;
   %SMASA_HOME%\lib\smclientclasses.jar;
   ```

4. Add %SMASA_CLASSPATH% to the beginning of the CLASSPATH definition.

   The modified CLASSPATH variable should resemble the following:

   ```
   set CLASSPATH=%SMASA_CLASSPATH%;%CLASSPATH%
   ```

5. Define the SM_JAVA_OPTIONS variable as follows:

   ```
   set SM_JAVA_OPTIONS= -Dsmasa.home=%SMASA_HOME%
   ```

6. Add %SM_JAVA_OPTIONS% to the execution entry.

   The modified execution entry should resemble the following:

   ```
   if "%WLS_REDIRECT_LOG%"=="" (
       echo Starting WLS with line:
       echo %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%
       %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%
   ) else (
       echo Redirecting output from WLS window to %WLS_REDIRECT_LOG%
       %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%  >"%WLS_REDIRECT_LOG%" 2>&1
   )
   ```

7. Save startWebLogic.cmd.
8. Restart the WebLogic Application Server for changes to take effect.

Set the WebLogic Environment for SiteMinder on UNIX

To set the environment for the SiteMinder Agent on a standalone WebLogic Server on UNIX, do the following:

- Define a Java environment variable SMASA.HOME that refers to the directory where the SiteMinder Agent for WebLogic is installed.
- Add the following SiteMinder Agent files and directories to the CLASSPATH variable:
  - *ASA_HOME*/conf
  - *ASA_HOME*/lib/smagentapi.jar
  - *ASA_HOME*/lib/smjavasdk2.jar
  - *ASA_HOME*/lib/sm_cryptoj.jar
  - *ASA_HOME*/lib/smclientclasses.jar
- Add the following SiteMinder Agent option to the WebLogic execution entry:
  - -Dsmasa.home=$SMASA_HOME

Example: Set the Environment for a Standalone WebLogic Server on UNIX

The following procedure is an example of how to set the environment for the SiteMinder Agent on a standalone WebLogic Server on UNIX.

1. Edit the startWebLogic.sh file.

   The startWebLogic.sh is located in *wl_install*/user_projects/domains/*your_domain/bin*

   where *wl_install* is the installed location of the WebLogic application server, and *your_domain* is the name of the WebLogic domain where the SiteMinder Agent is installed.

2. Define SMASA_HOME as follows:

   SMASA_HOME=*ASA_HOME*

   ASA_HOME

   Specifies the installed location of the SiteMinder Agent.

3. Define SMASA_CLASSPATH as follows:

   ```
   SMASA_CLASSPATH=$SMASA_HOME/conf:
   $SMASA_HOME/lib/smagentapi.jar;
   $SMASA_HOME/lib/smjavasdk2.jar;
   $SMASA_HOME/lib/sm_cryptoj.jar:
   $SMASA_HOME/lib/smclientclasses.jar:
   ```

4. Add ${SMASA_CLASSPATH} to the beginning of the CLASSPATH definition.

   The modified CLASSPATH variable should resemble the following:

   ```
   CLASSPATH=${SMASA_CLASSPATH}:${CLASSPATH}
   ```

5. Define the SM_JAVA_OPTIONS variable as follows:

   ```
   SM_JAVA_OPTIONS= -Dsmasa.home=$SMASA_HOME
   ```

6. 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}
   ```

7. Save startWebLogic.sh.
8. Restart the WebLogic Server.

Set Up the Agent Configuration File (WebAgent.conf)

The SiteMinder Agent installation creates an Agent configuration file (WebAgent.conf) that contains the following default configuration information:

- EnableWebAgent
- HostConfigFile
- AgentConfigObject

Note: For additional parameters that you can configure in the Agent configuration file, see Agent Configuration.

The Agent configuration file is located in the *ASA_HOME*\conf directory, where *ASA_HOME* is the location where you installed the SiteMinder Agent. For example:

- For Windows:

  C:\smwlsasa\conf

- For UNIX:

  /opt/smwlsasa/conf

You can use the default Agent configuration file for all of the SiteMinder Agent Providers, or you can create a separate configuration file for each Provider. The following table describes the features of both configurations:

Configuration	Features
Each Provider has a separate Agent configuration file	■ Agent configuration can be defined locally in the WebAgent.conf file so you can have different settings for each Provider. **Note:** The AllowLocalConfig parameter in the Agent Configuration Object must be set to yes. ■ Provider-specific information can be written to a separate log file. For example, you can configure one log file for Identity Asserter messages and a different log file for Authentication Provider messages. ■ You can enable or disable each Provider separately. ■ You can configure different cache settings for each Provider.
All Providers share the same configuration	■ Agent configuration is defined centrally in the Agent Configuration Object in the Policy Server and applies to all Providers. **Note:** The settings in the Agent Configuration Object are dynamic. You do not have to restart the Application Server for a setting change to take effect. ■ Information from all Providers is written to the same log. **Note:** You should not disable Providers if all Providers share the same configuration - WebLogic Server will not start if the SiteMinder Adjudication Provider is disabled.

Create an Agent Configuration File for Each SiteMinder Agent Provider

To create an Agent configuration file for each SiteMinder Agent Provider

1. In the Administrative UI:
   1. Open the Agent Configuration Object that you created for the SiteMinder Agent components.
   2. Set the AllowLocalConfig parameter to yes.
2. On the system where the SiteMinder Agent is installed:
   1. Create a configuration file by copying the WebAgent.conf file.

      The WebAgent.conf file is located in the *ASA_HOME*\conf directory.

Save the configuration file with a name that indicates the Provider to which the file applies. For example, name the configuration file for the Identity Asserter IAWebAgent.conf.

Be sure that the agentname, HostConfigFile, and the AgentConfigObject parameters are configured correctly.

b. Add parameters to the renamed WebAgent.conf file as described in Modify Configuration Files.

c. Repeat these steps for each SiteMinder Agent Provider.

3. Configure each SiteMinder Agent Provider to use the configuration file that you created for it when you configure the provider in the WebLogic Administrative Console.

More Information

Configure the SiteMinder Identity Asserter (see page 57)
Configure the SiteMinder Authentication Provider (see page 71)
Configure the SiteMinder Authorization Provider (see page 85)
Configure the SiteMinder Adjudication Provider (see page 91)

Reinstall the SiteMinder Agent

To reinstall the SiteMinder Agent, first uninstall it and then install it.

More information:

Uninstall the Agent (see page 53)
Install the SiteMinder Agent (see page 30)

Uninstall the Agent

The following sections contain information on uninstalling the Agent.

Uninstall the SiteMinder Agent from Windows

To uninstall the SiteMinder Agent from Windows

1. Open the WebLogic Server Administration Console
2. Remove configured SiteMinder Agent Identity Asserter, Authentication Provider and Authorization Provider modules (as applicable):
   a. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
   b. Click on the name of the realm you are configuring (for example, myrealm).
   c. Click the Providers tab.
   d. Click the tab for the type of provider that you are removing.
   e. For example, click the Authentication tab to remove the SiteMinder Identity Asserter.
   f. Select the SiteMinder provider from the list and click Delete.
   g. Click Yes to confirm the deletion.
3. Replace the SiteMinder Adjudication Provider (if configured):
   a. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
   b. Click on the name of the realm you are configuring (for example, myrealm).
   c. Click the Providers tab.
   d. Click the Adjudication tab.
   e. Select the SiteMinder Adjudication provider from the list and click Replace. The Create a New Adjudication Provider page opens.
   f. Specify a name for the new Adjudication Provider in the Name field. For example, AdjudicationProvider.
   g. Select a replacement Adjudication Provider from the Type drop-down list.
   h. Click OK to save the new Adjudication Provider.

   Important! Verify that there are other providers to assume the responsibility of the provider that you are removing *before* you restart the WebLogic Server.
4. Stop the WebLogic Server.

   Note: If you try to uninstall the SiteMinder Agent while WebLogic is still running, the SiteMinder Agent might not be fully uninstalled.

5. Open a command window and navigate to *ASA_HOME*\asa-wls-uninstall where you installed the SiteMinder Agent
6. Enter the following command and press Enter to start the uninstall:

   ```
   java -jar uninstaller.jar
   ```

   The uninstallation wizard appears.
7. Review the information in the Uninstall SiteMinder Agent dialog, then click Uninstall.
8. After confirmation indicates the uninstall is complete, click Done to exit.
9. If the uninstaller listed files it was not able to remove, delete them manually.
10. If necessary, manually delete the *ASA_HOME* directory (for example, smwlsasa) that the install program created.

Uninstall the SiteMinder Agent from UNIX

Before you uninstall, we recommend that you make copies of SiteMinder Agent configuration settings to have as a backup.

To uninstall the SiteMinder Agent from UNIX

1. Open the WebLogic Server Administration Console
2. Remove configured SiteMinder Agent Identity Asserter, Authentication Provider and Authorization Provider modules (as applicable):
   a. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
   b. Click on the name of the realm you are configuring (for example, myrealm).
   c. Click the Providers tab.
   d. Click the tab for the type of provider that you are removing.
   e. For example, click the Authentication tab to remove the SiteMinder Identity Asserter.
   f. Select the SiteMinder provider from the list and click Delete.
   g. Click Yes to confirm the deletion.
3. Replace the SiteMinder Adjudication Provider (if configured):
   a. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
   b. Click on the name of the realm you are configuring (for example, myrealm).
   c. Click the Providers tab.

d. Click the Adjudication tab.

e. Select the SiteMinder Adjudication provider from the list and click Replace. The Create a New Adjudication Provider page opens.

f. Specify a name for the new Adjudication Provider in the Name field. For example, AdjudicationProvider.

g. Select a replacement Adjudication Provider from the Type drop-down list.

h. Click OK to save the new Adjudication Provider.

Important! Verify that there are other providers to assume the responsibility of the provider that you are removing *before* you restart the WebLogic Server.

4. Stop the WebLogic Server.

   Note: If you try to uninstall the SiteMinder Agent while WebLogic is still running, the SiteMinder Agent might not be fully uninstalled.

   Verify that the PATH variable is set to the location of your JVM.

5. Open a UNIX shell and navigate to *ASA_HOME*/asa-wls-uninstall.

6. Enter one of the following commands and press Enter to start the uninstall:

   GUI mode

   ```
   java -jar uninstaller.jar
   ```

   Console mode

   ```
   java -jar uninstaller.jar -i console
   ```

   The uninstall program removes the SiteMinder Agent files.

7. Remove the *ASA_HOME* directory (for example, smwlsasa) that the installation created:

   a. Navigate to the directory one level above where the SiteMinder Agent is installed. For example:

      /opt

   b. Enter the following command and press Enter:

      ```
      rm -rf ASA_HOME
      ```

Chapter 3: Configure the SiteMinder Identity Asserter

This section contains the following topics:

Configure the SiteMinder Identity Asserter in SiteMinder

The following topics describe how to perform SiteMinder-side configuration of the SiteMinder Identity Asserter.

Create a SiteMinder Identity Asserter Validation Realm

Create a validation realm that allows the Identity Asserter to validate user credentials using session information received from SMSESSION cookies and X509 Client Certificates.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To configure the SiteMinder Identity Asserter validation realm

1. Open the SiteMinder Administrative UI.
2. Create an authentication scheme for the validation realm with the following properties:

   Name

   Specifies a unique name for the authentication scheme.

   Authentication Scheme Type

   An X509 Client Cert authentication scheme, such as X509 Client Cert Template, to enable the Identity Asserter to validate X.509 Client Certificates.

Server Name

The name of the server where WebLogic is installed.

Target

Leave the default value unchanged.

Note: This authentication scheme only passes credentials to the Policy Server for verification. It does not redirect requests to an SSL credential collector. Therefore, the Policy Server does not use the values specified in the Server Name and Target fields.

Note: See the Authentication Schemes chapter in the *SiteMinder Policy Server Configuration Guide* for instructions on creating an authentication scheme.

3. Create a domain and assign user directories that contain the users who can access the protected resources.

4. Create a realm with the following properties:

Domain

The domain you created in step 3.

Name

A unique name for the realm—for example, SiteMinder Identity Asserter Validation Realm.

Description

An optional description for the realm.

Agent

The name of the SiteMinder Agent identity that you created for the SiteMinder Agent.

Enter the Agent name in the text box or click the lookup button (...) to select the Agent name from a list of configured Agent identities.

Resource Filter

/smiavalidationrealm

Authentication Scheme

The authentication scheme you created in Step 2.

Maximum Timeout

This option must be disabled.

Idle Timeout

This option must be disabled.

Persistent Session

Non-persistent.

Note: If the session timeouts are not disabled, the identity assertion process might fail and the native WebLogic security services might challenge the request.

Note: You do not need to configure any rules for the Identity Asserter validation realm.

Configure the Identity Asserter to Only Handle Requests from SiteMinder Session Holders

To configure the SiteMinder Identity Asserter to handle only requests from users with valid SiteMinder session tickets or X.509 certificates (that is, not to challenge requests for credentials), verify that the ChallengeForCredentials Agent configuration parameter is disabled by setting it to NO in the associated Agent Configuration Object or Agent configuration file.

For example:

```
ChallengeforCredentials=NO
```

Configure the Identity Asserter to Challenge Requests for Credentials

To configure the SiteMinder Identity Asserter to challenge requests from users without an existing SiteMinder session ticket (and handle users that do have an existing SiteMinder session), perform the following steps:

- Set the ChallengeForCredentials Agent configuration parameter
- If using advanced authentication, synchronize overlapping settings between the IA and the Web Agent performing advanced authentication
- If using advanced authentication, configure the authentication scheme you want to use

When configured to challenge requests for credentials, the SiteMinder IA supports the following authentication schemes:

- Basic
- Basic over SSL
- HTML Forms
- X509 Client Certificate
- X509 Client Certificate and Basic
- X509 Client Certificate or Basic
- X509 Client Certificate and HTML Forms
- X509 Client Certificate or HTML Forms

Set the ChallengeForCredentials Parameter to Challenge Requests for Credentials

To configure the SiteMinder Identity Asserter to challenge requests from users without an existing SiteMinder session ticket and handle requests that do have an existing SiteMinder session, set the ChallengeForCredentials Agent configuration parameter to "YES" in the associated Agent Configuration Object or Agent configuration file.

For example:

```
ChallengeforCredentials=YES
```

Default is NO.

Synchronize Overlapping SiteMinder IA and Web Agent Configuration Parameters

When configured to challenge requests for credentials, for authentication schemes other than basic, the SiteMinder Identity Asserter redirects to a Web Agent to collect credentials. Verify that several Agent configuration parameters that apply to both Agent types have matching values to avoid related issues.

The fcccompatmode Agent configuration parameter handles backward compatibility of forms credential collection, which the SiteMinder IA does not support. Therefore, set this parameter to NO for both the SiteMinder IA and the Web Agent:

```
fcccompatmode="NO"
```

The SiteMinder IA does not support legacy encoding. Set the legacyencoding Agent configuration parameter to NO for both the SiteMinder IA and the Web Agent:

```
legacyencoding="NO"
```

The secureURLs setting in the Agent Configuration Object does not affect the fcccompatmode and legacyencoding parameters – the SiteMinder IA does not support them no matter what secureURLs is set to.

Note: The secureURLs parameter enables the Web Agent to encrypt all SiteMinder query parameters in a redirection URL. When this parameter is set to yes, the Agents encrypt query data when it returns an HTTP 302 status code (redirect response) to the browser. This functionality can be used when a requested resource is protected by an advanced authentication scheme. Use the Policy Server User Interface to set SecureURLs centrally in the Agent Configuration Object.

Additionally, the following parameters must match for both the SiteMinder IA and SiteMinder Web Agent if specified:

- EncryptAgentName
- IgnoreQueryData

Configure Authentication Schemes for Challenged Requests

If you are configuring the SiteMinder Identity Asserter to challenge requests for credentials using supported advanced (non-Basic) authentication schemes, configure the required authentication schemes, if they do not exist already. The following advanced authentication schemes are supported:

- Basic over SSL
- HTML Forms
- X509 Client Certificate
- X509 Client Certificate and Basic
- X509 Client Certificate or Basic
- X509 Client Certificate and HTML Forms
- X509 Client Certificate or HTML Forms

For more information, see the *SiteMinder Policy Server Configuration Guide*.

Create Realms for Challenged Requests

If your SiteMinder Identity Asserter is configured to challenge HTTP requests for credentials (the challengeforcredentials Agent configuration parameter is set to Yes), configure standard SiteMinder protection domains and realms to protect your Web container resources.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To create realms to protect your web applications

1. Open the SiteMinder Administrative UI.
2. Create a realm with the following properties:

   Domain

   The domain you created when you configured the SiteMinder Identity Asserter Validation Realm.

   Name

   A unique name for the realm—for example, Web App Protection Realm.

   Description

   An optional description. For example, "SiteMinder realm for validating/authenticating identities using the IA."

   Agent

   The name of the SiteMinder Agent identity that you created for the SiteMinder Agent for WebLogic.

   Enter the Agent name in the text box or click the lookup button (...) to select the Agent name from a list of configured Agent identities.

   Resource Filter

   /*web_app_context*

   web_app_context

   Specifies the J2EE Web application context for the protected web application.

   For example, /mywebapp.

   Authentication Scheme

   The authentication scheme you created for challenged requests.

   The SiteMinder IA handles Basic authentication itself; other authentication schemes are processed by a Web Agent.

More information:

Create a SiteMinder Identity Asserter Validation Realm (see page 57)
Configure Authentication Schemes for Challenged Requests (see page 61)

SiteMinder IA-Specific Agent Configuration Parameter Summary

Define the following Agent configuration parameters for the SiteMinder IA in an associated Agent Configuration Object or Agent configuration file.

Note: The SiteMinder Agent for Oracle WebLogic does not use the same agent configuration parameters as a SiteMinder Web Agent and even where parameters have similar names their values might not be compatible. Do not attempt to use the Agent Configuration Object for a SiteMinder Web Agent for the SiteMinder Agent for WebLogic. For a complete listing of configuration parameters for the SiteMinder Agent, see Agent Configuration Parameters (see page 125).

Required Parameter	Value	Description
AcceptTpCookie	**yes** or **no**	Configures the SiteMinder IA to assert identities from third-party SiteMinder session cookies generated using the SiteMinder SDK. For details, see "Enabling Single Sign-On" in the Agent API chapter of: ■ *CA SiteMinder Programming Guide for C* ■ *CA SiteMinder Programming Guide for Java* Default is NO. **Note:** If you configure the SiteMinder IA to accept third-party SiteMinder session cookies, also configure the SiteMinder Login Module to accept them so that it can assert WebSphere propagation tokens in situations when WebSphere must reestablish Subjects created by the SiteMinder IA.
ChallengeForCredentials	**yes** or **no**	Specifies whether the SiteMinder IA challenges for credentials. Default is NO.
AssertionAuthResource	String	If you are configuring the IA to *not* challenge requests for credentials, this value *must* match the value specified for the resource filter in the realm that you create for non-challenged requests. For example: assertionauthresource=/sitemindertai
CookieDomain	String	Name of the cookie domain. For example: cookiedomain="ca.com"

Required Parameter	Value	Description
		No default value. See also the cookiedomainscope parameter.
CookieDomainScope	Number	If specified, further defines the cookie domain for assertion of SiteMinder session cookies by the SiteMinder TAI. 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.
EncryptAgentName	**yes** or **no**	Specifies whether the agent name is encrypted when redirecting to the SiteMinder Web Agent for SiteMinder IA credential collection. *Must* match the value of the same parameter on the Web Agent responsible for advanced authentication. Default is NO.
FccCompatMode	**no**	Specifies whether to handle backward compatibility of forms credential collection, which the SiteMinder IA does not support. Therefore set this parameter to NO for *both* the SiteMinder IA *and* the Web Agent responsible for advanced authentication. For example: fcccompatmode="NO"
IgnoreQueryData	**yes** or **no**	Specifies whether the SiteMinder Agent will cache the entire URL (including the query strings) and send the entire URI to the Policy Server for rule processing. *Must* match the value of the same parameter on the Web Agent responsible for advanced authentication. Default is NO.
LegacyEncoding	**no**	Specifies whether to replace any dollar sign ($) characters in legacy URLs with a hyphen (-), which the SiteMinder IA does not support. Therefore set this parameter to NO for *both* the SiteMinder IA *and* the Web Agent responsible for advanced authentication. For example: legacyencoding="NO"
PersistentCookies	**yes** or **no**	Specifies whether the agent allows single sign-on for multiple browser sessions. When PersistentCookies is enabled, users who authenticate during one browser session will retain single sign-on capabilities for subsequent browser sessions. Default is NO.

Required Parameter	Value	Description
ServerErrorFile	String	Specifies a page to redirect a request to if a processing error is encountered. This can either be an HTTP or local file system resource. For example: servererrorfile="http://server.ca.com:88/errorpage.html" If this setting is not configured, a default message is output to the response when the IA encounters an error. The default message is "SiteMinder Agent encountered an error while handling request. Please ask the administrator to look for messages in the agent log to check for the cause."

Configure the SiteMinder Identity Asserter in WebLogic

The following topics contain information on configuring the SiteMinder Identity Asserter in WebLogic.

Configure the SiteMinder Identity Asserter in WebLogic

Configure the Identity Asserter in the Security Realms Node in the WebLogic Server Administration Console.

1. Start the WebLogic server and the WebLogic Server Administration Console.
2. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. If necessary, click the Authentication tab to display the Authentication Providers list.
6. (Optional) Delete the DefaultIdentityAsserter provider, if it is one of the authentication providers listed.
7. Click New to create a new Authentication Provider.

8. On the Create a New Authentication Provider page:
   - Specify a name for the Identity Asserter in the Name field. For example, SMIdentityAsserter
   - Select SiteMinderIdentityAsserter from the Type drop-down list.

   Note: If SiteMinderIdentityAsserter is not listed in the Type drop-down list, check the SiteMinder Agent installation to determine if it was successful.

9. Click OK to save the new Identity Asserter Provider.

10. Click the entry for your SiteMinder Identity Asserter in the Authentication Providers list to open it for editing:

    a. In the Active Types Chooser, Use the arrow key to move the SMSESSION and X.509 token types from the Available field to the Chosen field, as needed. Click Apply.

       Note: Each token type is handled by only one Identity Asserter. If you want the SiteMinder Identity Asserter to handle X.509 token types, be sure that no other Identity Asserter is configured to handle X.509 tokens.

    b. Click the Provider Specific subtab.

c. In the Config File field, enter the location of the configuration file for the SiteMinder Identity Asserter.

   If you are using the default Agent configuration file (WebAgent.conf), the location is *ASA_HOME*/conf/WebAgent.conf. If you created a new Agent configuration file for the Identity Asserter, be sure to enter the location and file name of the file you created.

   You can use an absolute or relative path. If you use a relative path, the configuration file will be relative to the smasa.home/conf or relative to your current WebLogic Server working directory, *WLS_HOME*/user_projects/*yourdomain*.

d. (Optional) In the User Name Attribute Mapper String field, specify an attribute in a user DN that stores a user name to be used *only* when the SiteMinder session cookie does not contain a NAME attribute.

   When the Identity Asserter receives a token that does not contain a NAME attribute through perimeter authentication, it extracts the user name from the specified attribute in the user DN and maps it to a user in the WebLogic user directory.

   For example, if the user DN is uid=jsmith, ou=myorganization, o=mycompany.com, and you specify uid in the User Name Attribute Mapper String field, the user name jsmith is passed to WebLogic.

11. Click Save.

12. If you have finished configuring SiteMinder Agent Providers, restart the WebLogic server for the changes to take effect.

    If you are configuring additional SiteMinder Agent Providers, you can restart the WebLogic server after all of the configuration steps are complete.

More Information

Install the SiteMinder Agent for WebLogic (see page 25)
Troubleshoot the SiteMinder Agent (see page 135)
Set Up the Agent Configuration File (WebAgent.conf) (see page 51)

Configure an Authentication Provider

For the Identity Asserter to propagate the user identity, an authentication provider must be able to verify that the user exists in a user store. You cannot use the Identity Asserter with the default authentication provider connected to the internal WebLogic LDAP; it is not supported. Configure the SiteMinder Authentication Provider, an authentication provider, or both for a directory supported by both CA and WebLogic.

Note: See the WebLogic documentation for information about configuring authentication providers other than the SiteMinder Authentication Provider.

Note: For a list of supported directories, see the SiteMinder support matrix on the Technical Support site.

More Information

Configure the SiteMinder Authentication Provider (see page 71)

Enable and Disabe the SiteMinder Identity Asserter

Enable the Identity Asserter after making all configuration changes, so that it can communicate with the Policy Server to gather management information. When you disable an Identity Asserter, it no longer validates the user and authentication defaults to the native WebLogic security mechanism.

You enable or disable the SiteMinder Identity Asserter in the Agent configuration file for the Agent.

1. Open the Agent configuration file in the *ASA_HOME*\conf directory.
2. Set the EnableWebAgent parameter as follows:
   - To enable the Identity Asserter, set EnableWebAgent to Yes as follows:

     EnableWebAgent="Yes"
   - To disable the Identity Asserter set EnableWebAgent to No:

     EnableWebAgent="No"

   Note: The EnableWebAgent parameter applies to all of the Providers that use the Agent configuration file. For example, if you configured all of the SiteMinder Agent Providers to use a single Agent configuration file, setting EnableWebAgent to yes enables all of the Providers.

More Information

Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Modify Configuration Files (see page 124)

Post-Configuration Notes

- To leverage an Identity Asserter, WebLogic requires that web applications are configured to use the CLIENT-CERT authentication method. For each web application, modify the deployment descriptor as follows:

  <auth-method>CLIENT-CERT</auth-method>

  Then redeploy the web application onto WebLogic server.
- Test your configuration and integration.
- Troubleshoot the configuration.

More Information

Troubleshoot the SiteMinder Agent (see page 135)

What to Do Next

To finish enabling the SiteMinder Agent solution to protect WebLogic resources, complete these steps in any order:

- Configure SiteMinder Authentication, Authorization, and Adjudication Providers to implement the complete SiteMinder Agent solution, as required.
- Configure SiteMinder policies.
- Verify that all configured SiteMinder Agent components are configured correctly.

More Information

Which SiteMinder Security Providers Do I Need? (see page 18)
Configure the SiteMinder Adjudication Provider (see page 119)
Configure the SiteMinder Authentication Provider (see page 71)
Configure the SiteMinder Authorization Provider (see page 85)
Configure Policies (see page 95)
Verify the SiteMinder Agent Installation and Configuration (see page 115)

Chapter 4: Configure the SiteMinder Authentication Provider

This section contains the following topics:

Overview

The SiteMinder Authentication Provider authenticates a user within the WebLogic security realm by checking the user's credentials against a SiteMinder directory.

After validating a user, the Authentication Provider adds the SiteMinder principal to the subject. The Authentication Provider can also obtain the groups that users belong to and populate the subject with a principal for each group.

For more information about how the SiteMinder Authentication provider works, see SiteMinder Authentication Provider. For more information about WebLogic principals and subjects and principals, see the WebLogic Documentation.

The following are the steps required to configure the SiteMinder Authentication provider.

1. Configure a SiteMinder realm for authentication.
2. Configure the SiteMinder Authentication provider in WebLogic.
3. Decide the following:
   a. To return physical or virtual group membership to WebLogic Server only.
   b. To return physical group membership to the WebLogic Server.
4. Enable the Authentication provider.
5. Configure an Authorization provider.
6. Create SiteMinder policies to protect WebLogic resources.
7. Verify that the SiteMinder Authentication Provider is configured correctly.

More Information

Configure the SiteMinder Authentication Provider in WebLogic (see page 73)
Enable and Disable the Authentication Provider (see page 83)
Configure the SiteMinder Authorization Provider (see page 85)
Configure Policies (see page 95)
Verify the SiteMinder Agent Installation and Configuration (see page 115)

Configure the SiteMinder Authentication Provider Realm

The SiteMinder Authentication Provider requires that you create an authentication realm using the Administrative UI. This realm allows the Authentication Provider to validate user credentials against a SiteMinder user directory.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To configure the SiteMinder Authentication Provider realm

1. Open the SiteMinder Administrative UI.
2. Create a domain and assign the user directories that contain users who are to access protected resources.
3. Create a realm with the following properties:

   Domain

   The domain you created in step 2.

   Name

   A unique name for the realm—for example, SiteMinder Authentication Provider Validation Realm.

   Description

   An optional description for the realm

   Agent

   The name of the SiteMinder Agent identity that you created for the SiteMinder Agent.

   Enter the Agent name in the text box or click Lookup to select the Agent name from a list of configured Agent identities.

   Resource Filter

   /smauthenticationrealm

Authentication Scheme

Basic.

Maximum Timeout

This option must be disabled.

Idle Timeout

This option must be disabled.

Persistent Session

Non-persistent.

You do not need to configure any rules or policies for the Authentication provider validation realm.

Important! After initial setup, do not attempt to reconfigure the SiteMinder Authentication validation realm while the WebLogic Server is running.

Configure the SiteMinder Authentication Provider in WebLogic

The following sections discuss configuring the SiteMinder authentication provider in WebLogic.

Configure the SiteMinder Authentication Provider

Configure the Authentication Provider in the Security Realms Node in the WebLogic Administration console.

To configure the SiteMinder Authentication Provider

1. Start the WebLogic Server and the WebLogic Server Administration Console.
2. In the navigation frame on the left of the Console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. If necessary, click the Authentication tab to display the Authentication Providers list.
6. Click New to create a new Authentication Provider.

7. On the Create a New Authentication Provider page:

   - Specify a name for the Authentication Provider in the Name field. For example, SMAuthenticationProvider.
   - Select SiteMinderAuthenticationProvider from the Type drop-down list.

   Note: If SiteMinderAuthenticationProvider is not listed, check the SiteMinder Agent installation to determine if it was successful.

8. Click OK to save the new Authentication Provider.

9. Click the entry for your SiteMinder Authentication Provider in the Authentication Providers list to open it for editing.

10. In the SiteMinder Authentication Provider settings page, complete the following:

    a. In the Control Flag field, select the priority that applies to the SiteMinder Authentication Provider.

       Note: If your environment includes other authentication providers, we recommend setting the Control Flag for the SiteMinder Authentication Provider to SUFFICIENT.

    b. Click the Provider Specific tab.

    c. In the SMAuth Provider Config File field, enter the location of the configuration file for the Authentication Provider.

       If you are using the default Agent configuration file, the location is *ASA_HOME*/conf/WebAgent.conf. If you created a new Agent configuration file for the Authentication Provider, be sure to enter the location and file name of the file you created.

       You can use an absolute or relative path. If you use a relative path, the configuration file will be relative to the directory smasa.home/conf or relative to your current WebLogic Server working directory, *WLS_HOME*/user_projects/*yourdomain*.

    d. Click Save.

11. If multiple authentication providers are configured for the security realm, specify the order in which WebLogic executes the authentication providers as described in Configure the Execution Order.

12. If the Default Authentication Provider is configured for the security realm, change the Control Flag setting for the Default Authentication Provider from REQUIRED to SUFFICIENT.

13. Enable the Authentication Provider.

14. Enable SiteMinder logging.

15. Restart the WebLogic server and check SiteMinder logs to verify that the Authentication Provider is configured correctly.

    If you are configuring additional SiteMinder Agent SiteMinder Agent Providers, you can restart the WebLogic server after all of the configuration steps are complete.

More Information

Install the SiteMinder Agent for WebLogic (see page 25)
Troubleshoot the SiteMinder Agent (see page 135)
Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Logging (see page 107)
Enable and Disable the Authentication Provider (see page 83)
Set the Control Flag (see page 76)

Determine How Users Are Authenticated

In a WebLogic security realm that includes multiple authentication providers, the process for authenticating users is determined by the following:

- The execution order of the configured authentication providers
- The Control Flag setting for each authentication provider

Configure the Execution Order

You can list the order in which WebLogic executes authentication providers in the WebLogic Server Administration Console.

When a user attempts to access a protected resource, WebLogic executes the first authentication provider in the list. After the first authentication attempt, WebLogic determines whether to execute the next authentication provider based on the following criteria:

- The outcome of the first authentication attempt
- The control flag setting for the authentication provider that performed the authentication

For example, if the SiteMinder Authentication Provider is configured first in the execution order with control flag setting SUFFICIENT and it fails to authenticate a user, the user request is rejected immediately. WebLogic does not execute any other Authentication Providers (unless other providers are set to REQUIRED).

To configure the execution order:

1. Start the WebLogic server and the WebLogic Server Administration Console.
2. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. If necessary, click the Authentication tab to display the Authentication Providers list.
6. Click Reorder.
7. In the Reorder Authentication Providers list box, select a configured provider and use the arrows to change its position in the list.
8. Click OK.

More Information

Set the Control Flag (see page 76)

Set the Control Flag

When you configure an authentication provider in the WebLogic Administrative Console, you set the control flag on the General tab on the properties page for the provider.

The Control Flag determines how much weight an authentication decision has in an environment that includes multiple Authentication Providers. You can select one the following options for the control flag:

REQUIRED	This Authentication provider is always called, and the user must always pass its authentication test. After this authentication provider attempts to authenticate the user, WebLogic executes the other configured authentication providers, regardless of whether the authentication attempt succeeded.
REQUISITE	The authentication provider must authenticate the user. After the user is authenticated by the authentication provider, other authentication providers attempt to validate the user. The user can fail to authenticate through any other authentication provider, except providers that have the control flag set to REQUIRED.

SUFFICIENT	If a user is authenticated by the authentication provider, no other authentication is required (unless another authentication provider has the control flag set to REQUIRED). REQUIRED modules listed after a module flagged SUFFICIENT do not run if it passes.
OPTIONAL	The user can pass or fail the authentication provider authentication. If all of the authentication providers are set to OPTIONAL, the user must pass at least one authentication test.

See the WebLogic documentation for more information about the control flag.

Configure the Agent to Return Group Membership to WebLogic Using Responses

During user authentication, the SiteMinder Agent can return physical or virtual group membership information to the WebLogic Server by using SiteMinder HTTP header responses from the Policy Server. When the SiteMinder Agent receives responses containing the _SM_WLS_GROUP=*group name* syntax (where *group_name* is a response attribute value from the Policy Server that could be a physical group name from the user store or a virtual group), the SiteMinder Agent converts the *group_name* value to a group principal and adds this principal to the subject after successful authentication. The SiteMinder Agent adds the same amount of group principals as responses received from the Policy Server.

Example: Configure Groups as Responses for the SiteMinder Agent

Note: The following procedures provide an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To configure groups as responses

1. In the SiteMinder Authentication Realm, configure an OnAuthAccept rule named Group Authentication Rule with a ***** resource filter.
2. In the policy domain for the SiteMinder Authentication Realm, create SiteMinder responses with a static HTTP header attribute for the following sample WebLogic groups:

Name	Attribute Kind	Variable Name	Variable Value
Group Administrators	Static HTTP Header	_SM_WLS_GROUP	Administrators
Group Deployers	Static HTTP Header	_SM_WLS_GROUP	Deployers
Group Monitors	Static HTTP Header	_SM_WLS_GROUP	Monitors
Group Operators	Static HTTP Header	_SM_WLS_GROUP	Operators

3. In the policy domain for the SiteMinder Authentication Realm:
   a. Configure a policy named **Group Administrator Policy**.
   b. Attach the Administrator group or users, who belong to the Administrator group, to this policy.
   c. Attach the Group Authentication Rule to this policy.
   d. Bind the Group Administrator response to this rule.
   e. Repeat this step and configure separate policies for the Deployers, Operators, and Monitors groups.
   f. Bind the Group Administrator response to this rule.
   g. Repeat this step and configure separate policies for the Deployers, Operators, and Monitors groups.
   h. Repeat this step and configure separate policies for the Deployers, Operators, and Monitors groups.

Configure the Agent to Return Group Membership to WebLogic Server Using Agent Configuration Parameters

By acting as an Authentication Provider for WebLogic Server, the SiteMinder Agent provides the user-group relationships needed during authentication.

To configure the Agent to return a user's physical group name from the user store to the WebLogic Server, modify the Agent Configuration Object with:

- Agent configuration parameters
- SiteMinder administrator and user store parameters

One parameter, SmUserDirectory, differs if the user store is associated with a CA Identity Manager (licensed separately) or non-CA Identity Manager environment. A CA Identity Manager environment is a view of a management namespace that allows CA Identity Manager administrators to manage objects—users, groups, and organizations—with a set of associated roles and tasks.

Note: For more information, see the CA Identity Manager documentation.

Note: The SiteMinder Agent only supports LDAP-based user directories and not those residing in a relational database.

SiteMinder User Directory Configured in Identity Manager Environment

In this set up, you must have CA Identity Manager (licensed separately) installed and configured. The Policy Server user store must be associated with a CA Identity Manager environment. Then, configure the SMAdminUserName, SMAdminUserPassword, and SmUserDirectory parameters in the SiteMinder Agent Configuration Object.

Note: Verify that the CA Identity Manager smjavasdk2.jar library is included in the classpath; the SiteMinder Agent uses it to query CA Identity Manager.

We recommend that you set these parameters centrally in the Agent Configuration Object using the Administrative UI because the SiteMinder administrator password cannot be encrypted in the WebAgent.conf file. Using this interface, you can encrypt this password in the Agent Configuration Object stored in the policy store.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To modify the Agent Configuration Object in the Policy Store

1. Open the SiteMinder Administrative UI.
2. Open the Agent Configuration Object that you want to modify.
3. Add the following SiteMinder administrator parameters:

Parameter Name	Value	Description
SMAdminUserName	SiteMinder administrator user name	User name of the Administrator with full permissions to manage all SiteMinder domain objects and users.
SMAdminUserPassword	Encrypted password	Encrypted Administrator password

4. Add the following CA Identity Manager environment parameter:

Parameter Name	Value	Description
SMUserDirectory	IMS, *IMS_environ*	(IMS means CA Identity Manager.) *IMS_environ* is the name of the CA Identity Manager environment. For example: IdentityManagerEnv

Note: Because SmUserDirectory can be a multivalued parameter, you can configure more than one user directory in the Agent Configuration Object. You can use multiple parameters to declare more than one DMS configuration or CA Identity Manager environment.

5. Restart the WebLogic Server for configuration changes to take effect. Reboot this server because the SmUserDirectory parameter is not dynamic.

SiteMinder User Directory Not Configured in Identity Manager Environment (Use DMS API)

In this set up, the SiteMinder Agent requires the SiteMinder Administrator credentials and the SiteMinder user directory structure configured within the Authentication Realm to gather user and group information. In the Agent Configuration Object, you set the credentials using the SMAdminUserName and SMAdminUserPassword parameters and configure this structure using the SmUserDirectory multivalued parameter.

We recommend that you set these parameters centrally in the Agent Configuration Object using the Administrative UI because the SiteMinder administrator password cannot be encrypted in the WebAgent.conf file. Using this interface, you can encrypt this password in the Agent Configuration Object stored in the policy store.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To modify the Agent Configuration Object in the Policy Store

1. Open the SiteMinder Administrative UI.
2. Open the Agent Configuration Object that you want to modify.
3. Add the following SiteMinder administrator parameters:

Parameter Name	Value	Description
SMAdminUserName	SiteMinder administrator user name	User name of the Administrator with full permissions to manage all SiteMinder domain objects and users.
SMAdminUserPassword	Encrypted password	The encrypted administrator password

4. Add an "SmUserDirectory" parameter with the following comma-separated values:

   Directory_Type, *LDAP_User_Directory_Name*, *usernameobjectclass*, *username-field*, *username-description*, *groupname-objectclass*, *groupname-attribute*, *groupname-description*

Variable	Description	Example Value
Directory_Type	Type of directory, which could be either DMS or IMS. Specify DMS if you are not using CA Identity Manager.	DMS
LDAP_User_Directory_Name	Name of the SiteMinder user directory configured in the Policy Server.	MyLDAPUserDirectory
username-objectclass	User object class.	inetorgperson^person
username-field	User name field.	uid
username-description	User name description.	description
groupname-objectclass	Group name object class.	groupofuniquenames
groupname-attribute	Group name attribute.	cn
groupname-description	Group name description.	description

Important! Values containing a comma can be concatenated using the ^ symbol. For example, inetorgperson, person becomes inetorgperson^person as in the following sample Value field entry: DMS, MyLDAPUserDirectory, inetorgperson^person, uid, description, groupofuniquenames, cn, description

Note: Because SmUserDirectory can be a multivalued parameter, you can configure more than one user directory in the Agent Configuration Object. You can use multiple parameters to declare more than one DMS configuration or CA Identity Manager environment.

5. In the startWebLogic.cmd file (for Windows) or startWebLogic.sh file (for UNIX), add the path of the smjavasdk2.jar file (located in *ASA_HOME*\lib) file to the WebLogic Server CLASSPATH.

   The modified CLASSPATH variable should resemble the following on Windows:

   ```
   set SMASA_CLASSPATH=%ASA_HOME%\conf;
   %ASA_HOME%\lib\smagentapi.jar;%ASA_HOME%\lib\sm_cryptoj.jar;
   %ASA_HOME%\lib\smclientclasses.jar;%ASA_HOME%\lib\smjavasdk2
   .jar;
   set
   CLASSPATH=%SMASA_CLASSPATH%;%WEBLOGIC_CLASSPATH%;
   %POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\rt.jar;
   %WL_HOME%\server\lib\webservices.jar;%CLASSPATH%
   ```

   The modified CLASSPATH variable should resemble the following on UNIX:

   ```
   SMASA_CLASSPATH=$ASA_HOME\conf;
   $ASA_HOME\lib\smagentapi.jar;$ASA_HOME\lib\sm_cryptoj.jar;
   $ASA_HOME\lib\smclientclasses.jar;$ASA_HOME\lib\smjavasdk2.jar;
   CLASSPATH=$SMASA_CLASSPATH%;$WEBLOGIC_CLASSPATH;
   $POINTBASE_CLASSPATH%;$JAVA_HOME\jre\lib\rt.jar;
   $WL_HOME\server\lib\webservices.jar;$CLASSPATH
   ```

6. Restart the WebLogic Server for configuration changes to take effect. Reboot this server because the SmUserDirectory parameter is not dynamic.

Enable and Disable the Authentication Provider

After you set up the Authentication Provider realm and configure the Authentication Provider in the WebLogic administration console, enable the SiteMinder Authentication Provider so that it can authenticate users against SiteMinder user directories.

When you disable a SiteMinder Authentication Provider, it no longer authenticates users and authentication defaults to other configured authentication providers.

You enable or disable the SiteMinder Authentication Provider in the Agent configuration file for the Agent.

Note: If you are using a single Agent configuration file for multiple SiteMinder Agent providers including the Authentication Provider and you have already enabled a Provider in that file, you do not need to complete this procedure. Continue the configuration process by completing the verification steps.

To enable or disable the SiteMinder Authentication Provider

1. Open the Agent configuration file in the *ASA_HOME*\conf directory.
2. Set the EnableWebAgent parameter as follows:
   - To enable the Authentication Provider, set EnableWebAgent="Yes"
   - To disable the Authentication Provider, set EnableWebAgent="No"

   Note: The EnableWebAgent parameter applies to all of the Providers that use the Agent configuration file. For example, if you configured all of the SiteMinder Agent Providers to use a single Agent configuration file, setting EnableWebAgent to yes, enables all of the Providers.

More Information

Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Verify the SiteMinder Agent Installation and Configuration (see page 115)
Modify Configuration Files (see page 124)

Chapter 5: Configure the SiteMinder Authorization Provider

This section contains the following topics:

Overview

After a user has been authenticated by the SiteMinder Authentication Provider, the SiteMinder Authorization Provider evaluates SiteMinder policies to determine whether the user can access a protected WebLogic resource.

Note: The SiteMinder Authorization Provider only authorizes requests that have been authenticated by the SiteMinder Authentication Provider.

To configure the SiteMinder Authorization Provider.

1. Configure a SiteMinder realm for authorization.
2. Configure the SiteMinder Authorization Provider in WebLogic.
3. Enable the Authorization provider.
4. Configure SiteMinder policies.
5. Configure the SiteMinder Adjudication Provider.
6. Verify that the Authorization Provider is configured correctly.

More Information

SiteMinder Authentication Provider (see page 15)
SiteMinder Authorization Provider (see page 16)
Enable and Disable the Authorization Provider (see page 89)
Configure Policies (see page 95)
Configure the SiteMinder Adjudication Provider (see page 91)
Verify the SiteMinder Agent Installation and Configuration (see page 115)

Configure the SiteMinder Authorization Provider Realm

To enable granular policy definition for WebLogic resources, the SiteMinder Authorization Provider requires that you create a realm in the Administrative UI. This realm allows you to create rules and policies that determine whether a user is allowed to access a protected WebLogic resource.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To configure a realm for the SiteMinder Authorization Provider

1. Open the SiteMinder Administrative UI.
2. Create a realm with the following properties:

   Domain

   A domain such as the domain you created for the SiteMinder Authentication Provider realm.

   Name

   A unique name for the realm (for example, SiteMinder Authorization Provider Realm).

   Description

   An optional description for the realm

   Agent

   The name of the SiteMinder Agent identity that you created for the SiteMinder Agent.

   Enter the Agent name in the text box or click Lookup to select the Agent name from a list of configured Agent identities.

   Resource Filter

   /wlsspiaz

   Authentication Scheme

   Basic.

   If you are using the SiteMinder Authorization Provider with the SiteMinder IA, the protection level for the authentication scheme for the Authorization Provider must be the same or lower than the protection level for realms protected by the front-end Web Agent. If the protection level is higher, the Authorization Provider rejects the user using the WebLogic native security services.

Maximum Timeout

An appropriate session timeout value.

Note: Bear in mind that there is no session synchronization between the SiteMinder Agent and WebLogic when setting timeout values.

Idle Timeout

An appropriate session idle timeout value.

Note: Bear in mind that there is no session synchronization between the SiteMinder Agent and WebLogic when setting timeout values.

Persistent Session

Non-persistent.

Configure the SiteMinder Authorization Provider in WebLogic

Configure the Authorization Provider in the Security Realms Node in the WebLogic Administration console.

To configure the Authorization Provider in WebLogic

1. Open the WebLogic Server Administration Console.
2. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. Click the Authorization tab to display the Authorization Providers list.
6. Click New to create a new Authorization Provider.
7. On the Create a New Authorization Provider page:

   a. Specify a name for the Authorization Provider in the Name field. For example, SMAuthorizationProvider.

   b. Select SiteMinderAuthorizationProvider from the Type drop-down list.

   Note: If SiteMinderAuthorizationProvider is not listed, check the SiteMinder Agent installation to determine if it was successful.

8. Click OK to save the new Authorization Provider.

9. Click the entry for your SiteMinder Authorization Provider in the Authentication Providers list to open it for editing:

   a. Click the Provider Specific tab.

   b. To determine what access decision the SiteMinder Authorization provider returns when the requested resource is not authenticated by SiteMinder, set the Abstain if Not Authenticated flag as follows:

If the Flag is...	**The result from the SiteMinder Authorization Provider is...**
Enabled	ABSTAIN
Disabled	DENY

   The effect that these access decisions have on a user's access to a WebLogic resource depends on how the Adjudication Provider is configured.

   c. To determine what access decision the SiteMinder Authorization provider returns when the requested resource is not protected by a SiteMinder policy, set the Abstain if Not Protected flag as follows:

If the Flag is...	**The result from the SiteMinder Authorization Provider is...**
Enabled	ABSTAIN
Disabled	PERMIT

   d. In the SMAz Provider Config File field, enter the location of the configuration file for Authorization Provider.

   If you are using the default Agent configuration file, the location is *ASA_HOME*/conf/WebAgent.conf. If you created a new Agent configuration file for the Authorization Provider, be sure to enter the location and file name of the file you created.

   You can use an absolute or relative path. If you use a relative path, the configuration file will be relative to the directory smasa.home/conf or relative to your current WebLogic Server working directory, *WLS_HOME*/user_projects/*yourdomain*.

10. Click Save.

11. Enable the Authorization Provider.
12. Enable SiteMinder logging.
13. If you finished configuring SiteMinder Agent Providers, restart the WebLogic server and check SiteMinder logs to verify that the Authorization Provider is configured correctly.

    If you are configuring additional SiteMinder Agent Providers, you can restart the WebLogic server after all of the configuration steps are complete.

More Information

Install the SiteMinder Agent for WebLogic (see page 25)
SiteMinder Agent Installation and Configuration Files (see page 123)
SiteMinder Adjudication Provider (see page 17)
Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Enable and Disable the Authorization Provider (see page 89)
Logging (see page 107)

Enable and Disable the Authorization Provider

After you set up the Authorization Provider realm and configure the Authorization Provider in the WebLogic Server Administration Console, enable the SiteMinder Authorization Provider so that it can authorize users against SiteMinder user directories.

When you disable a SiteMinder Authorization Provider, it no longer authorizes users and authorization defaults to other configured authorization providers.

Important! If the SiteMinder Authorization Provider is the only authorization provider configured for a security realm and you disable it, all authorization requests will be denied.

To enable or disable the SiteMinder Authorization Provider in the Agent configuration file for the Agent

Note: If you are using a single Agent configuration file for multiple SiteMinder Agent providers including the Authorization Provider and you have already enabled a Provider in that file, you do not need to complete this procedure. Continue the configuration process by completing the verification steps.

1. Open the Agent configuration file in the *ASA_HOME*/conf directory.
2. Set the EnableWebAgent parameter as follows:
   - To enable the Authorization Provider, set EnableWebAgent to Yes as follows:

     EnableWebAgent="Yes"
   - To disable the Authorization Provider, set EnableWebAgent to No:

     EnableWebAgent="No"

   Note: The EnableWebAgent parameter applies to all of the Providers that use the Agent configuration file. For example, if you configured all of the SiteMinder Agent Providers to use a single Agent configuration file, setting EnableWebAgent to yes, enables all of the Providers.

More Information

Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Modify Configuration Files (see page 124)

Chapter 6: Configure the SiteMinder Adjudication Provider

This section contains the following topics:

Overview

The SiteMinder Adjudication Provider makes a final access decision after a user has been authenticated and authorized for a protected resource. In an environment that uses more than one authorization provider, the SiteMinder Adjudication Provider resolves conflicts by weighing the result the access decision of each authorization provider. See SiteMinder Adjudication Provider for more information.

Note: A WebLogic security domain can have only one Adjudication Provider.

To configure the SiteMinder Adjudication Provider.

1. Configure the SiteMinder Adjudication Provider in WebLogic.
2. Enable the Adjudication Provider.
3. Validate that the Adjudication Provider is configured correctly.

More Information

Enable and Disable the Adjudication Provider (see page 93)
Verify the SiteMinder Agent Installation and Configuration (see page 115)

Configure the SiteMinder Adjudication Provider in WebLogic

Configure the SiteMinder Adjudication Provider in the Security Realms Node in the WebLogic Server Administration Console.

To configure the SiteMinder Adjudication Provider in WebLogic

1. Open the WebLogic Server Administration Console.
2. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. Click the Adjudication tab to display the Adjudication Providers list.
6. Click Replace to replace the default adjudication provider.
7. On the Create a New Adjudication Provider page:

   a. Specify a name for the Adjudication Provider in the Name field. For example, SMAdjudicationProvider.

   b. Select SiteMinderAdjudicationProvider from the Type drop-down list.

   Note: If SiteMinderAdjudicationProvider is not listed, check the SiteMinder Agent installation to determine if it was successful.

8. Click OK to save the new Adjudication Provider.
9. Click the entry for your SiteMinder Adjudication Provider to open it for editing:

   a. Click the Provider Specific tab.

   b. In the SMAdjudication Provider Config File field, enter the location of the configuration file for the Adjudication Provider.

      If you are using the default Agent configuration file, the location is *ASA_HOME*/conf/WebAgent.conf. If you created a new Agent configuration file for the Adjudication Provider, be sure to enter the location and file name of the file you created.

      You can use an absolute or relative path. If you use a relative path, the configuration file will be relative to the directory smasa.home/conf or relative to your current WebLogic Server working directory, *WLS_HOME*/user_projects/*yourdomain*.

c. In the SiteMinder Permission Decision field, select one of the following options:

SiteMinder Precedence	The SiteMinder Adjudication Provider assigns maximum weight to the result returned by a SiteMinder Authorization Provider.
Equal Precedence	The SiteMinder Adjudication Provider assigns equal weight to the results returned by all configured Authorization Providers in the security realm.

10. Click Save.
11. Enable the Adjudication Provider.
12. Configure logging.
13. Restart the WebLogic Server and check SiteMinder logs to verify that the Adjudication Provider is configured correctly.

More Information

Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Install the SiteMinder Agent for WebLogic (see page 25)
SiteMinder Adjudication Provider (see page 17)
Enable and Disable the Adjudication Provider (see page 93)
Logging (see page 107)

Enable and Disable the Adjudication Provider

After you configure the SiteMinder Adjudication Provider in the WebLogic administration console, enable the Adjudication Provider so that it can evaluate authorization decisions.

Important! If the SiteMinder Adjudication Provider is disabled, it will always return DENY access to any resource, including the WebLogic administration functions; you will not be able to start the WebLogic server or access the WebLogic Administration Console.

You enable or disable the SiteMinder Adjudication Provider in the configuration file (WebAgent.conf) for the Agent.

Note: If you are using a single Agent configuration file for multiple SiteMinder Agent providers including the Adjudication Provider and you have already enabled a Provider in that file, you do not need to complete this procedure. Continue the configuration process by completing the verification steps.

To enable and disable the SiteMinder Adjudication Provider

1. Open the Agent configuration file in the *ASA_HOME*/conf directory.
2. Set the EnableWebAgent parameter as follows:

    To enable the Adjudication Provider, set EnableWebAgent="Yes"

    To disable the Adjudication Provider, set EnableWebAgent="No"

    Note: The EnableWebAgent parameter applies to all of the Providers that use the Agent configuration file. For example, if you configured all of the SiteMinder Agent Providers to use a single Agent configuration file, setting EnableWebAgent to yes enables all of the Providers.

More Information

Set Up the Agent Configuration File (WebAgent.conf) (see page 51)
Verify the SiteMinder Agent Installation and Configuration (see page 115)
Modify Configuration Files (see page 124)

Chapter 7: Configure Policies

This section contains the following topics:

Configure Policies for the SiteMinder Authorization Provider

To configure the SiteMinder Agent for WebLogic to protect WebLogic resources using the SiteMinder Authorization Provider, create policies in a similar manner as you would to protect a web resource with a Web Agent. The only differences are:

- All rules in your policies must be created within the SiteMinder Authorization Provider validation realm and use SiteMinder to WebLogic resource mapping conventions to specify the WebLogic resources that you want to protect.
- Rules for non-URL resources must specify the Web Agent **Get** rule action. Rules for URL resources can specify **Post** and other HTTP-based actions.
- Responses, variables and policy expressions are not supported.

Note: For complete information about SiteMinder policy configuration, see the *SiteMinder Policy Server Configuration Guide.*

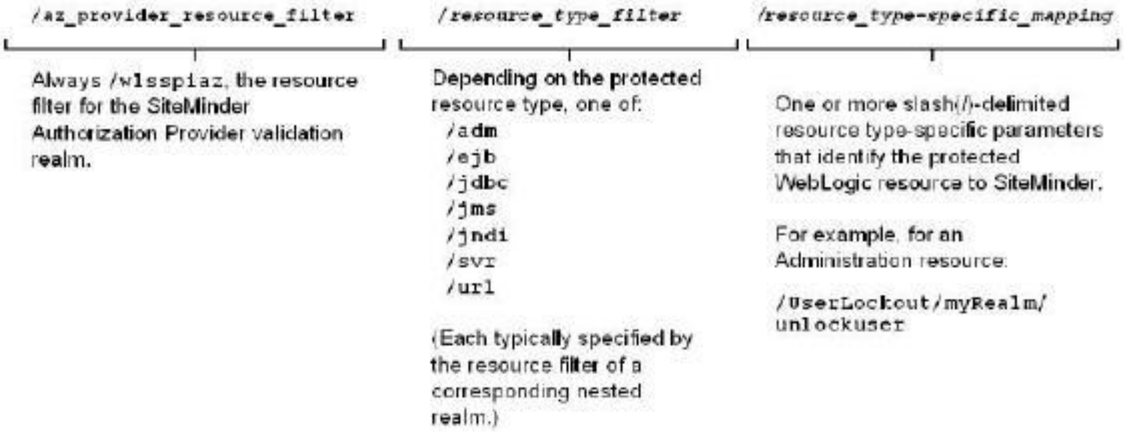
More Information

SiteMinder Resource Mapping for WebLogic Resources (see page 95)

SiteMinder Resource Mapping for WebLogic Resources

The Resource field in a SiteMinder rule specifies the resource that is the subject of the rule. The complete resource specification (shown by the Effective Resource field on the Rule dialog) is a concatenation of the values of the Resource Filter of the parent realm (or realms in a nested realm environment) and the Resource field of the rule itself. Resources that are not accessed through a URL must be defined using special mapping conventions.

This section describes the SiteMinder resource mapping for WebLogic resources. This mapping, which is summarized in the following figure, provides a means of representing WebLogic resources in the realms and rules that make up your authorization policies.



Always /wlsspiaz, the resource filter for the SiteMinder Authorization Provider validation realm.

Depending on the protected resource type, one of:
/adm
/ejb
/jdbc
/jms
/jndi
/svr
/url

(Each typically specified by the resource filter of a corresponding nested realm.)

One or more slash(/)-delimited resource type-specific parameters that identify the protected WebLogic resource to SiteMinder.

For example, for an Administration resource:

/UserLockout/myRealm/unlockuser

Configure the az_provider_resource_filter Section

The first section of the mapping, az_provider_resource_filter, tells SiteMinder that the resource is a WebLogic resource protected by the SiteMinder Authorization Provider. Its value is static (/wlsspiaz) and is defined by the resource filter in the Authorization Provider validation realm.

Configure the resource_type_filter Section

The second section of the mapping, *resource_type_filter*, tells SiteMinder what type of WebLogic resource is protected. Its value is determined by the type of resource, as shown in the following table.

Resource Type	***resource_type_filter* value**
Administration Resource	adm
EJB Resource	ejb
JDBC Resource	jdbc
JMS Resource	jms
JNDI Resource	jndi
Server Resource	svr
URL Resource	url

Note: If the SiteMinder Resource Mapper obtains a resource from WebLogic that is not of the types shown in the previous table, the default resource mapping is a concatenation of the requested resource values obtained from the WebLogic resource type. Spaces within the resource values are converted to a slash (/). You can use debug log messages from the SiteMinder Authorization Provider to obtain information about the requested WebLogic resource and the SiteMinder mapping of the WebLogic resource to a SiteMinder resource.

We recommend that you configure a nested realm under the SiteMinder Authorization Provider validation realm for each WebLogic resource type, specifying the appropriate *resource_type_filter* as the resource filter as shown in the following table.

Nested Realm Resource Filter	Nested Realm Type	Realm Contents
/adm	Administration Resource realm	Rules for Administration Resources
/jdbc	JDBC resource realm	Rules for JDBC resources
/jms	JMS resource realm	Rules for JMS resources
/jndi	JNDI resource realm	Rules for JNDI resources
/svr	Server resource realm	Rules for Server resources
/url	URL resource realm	Rules for URL resources

Note: If you implement your security policies using nested realms, verify that the Enable Nested Security setting is enabled on the SiteMinder Global Settings dialog. Additionally, create a simple *allow access* rule in the SiteMinder Authorization Provider validation realm and include it in your authorization policy. For more information about nested realms, see the *SiteMinder Policy Server Configuration Guide*.

Alternatively, include the *resource_type_filter* value as part of the resource specification in the rule.

Configure the resource_type-specific_mapping Section

The final section of the mapping, *resource_type-specific_mapping*, tells SiteMinder the specifics of the protected resource. Its value is one or more slash(/)-delimited parameters specific to the type of resource being protected (as defined in the *resource_type_filter* section).

Administration Resources

To protect a WebLogic Administration resource, *resource_type_filter* must specify the following parameters (in the order shown):
/category/realm/action

Parameter Name	Description	Field value example
category	Category associated with the administration resource.	UserLockout
realm	Name of the WebLogic security realm.	MyRealm
action	Action associated with the resource. Optional — if not specified, defaults to "GET".	unlockuser

For example, for an Administration Resource with the following properties:

category=UserLockout, realm=myrealm, action=unlockuser

The complete resource mapping (effective resource) would be:
/wlsspiaz/adm/UserLockout/myRealm/unlockuser

EJB Resources

To protect a WebLogic EJB resource, *resource_type_filter* must specify the following parameters (in the order shown):
/app/module/ejb/method/methodInterface/methodParams

Parameter Name	Description	Field value examples
app	Name of the application containing the EJB	MyApp
module	Name of the module containing the EJB.	MyJarFile
ejb	Name of the EJB	myEJB
method	Method executed on the EJB	myMethod
methodInterface	Method interface invoked on the EJB	Home

Parameter Name	Description	Field value examples
methodParams	Arguments in the signature of the EJB method. Treat multiple arguments as separate slash(/)-delimited parameters.	java.lang.String, int

For example, for an EJB application with the following properties:

app=myApp, module=MyJarFile, ejb=myEJB, method=myMethod, methodInterface=Home, methodParams=(java.lang.String, int)

The complete resource mapping (effective resource) would be:
/wlsspiaz/ejb/myApp/MyJarFile/myEJB/myMethod/Home/java.lang.String/int

JDBC Resource

To protect a WebLogic JDBC resource, *resource_type_filter* must specify the following parameters (in the order shown):
/category/resource/action

Parameter Name	Description	Field value examples
category	Category of the JDBC resource.	connectionPool
resource	Name of the JDBC resource.	myPool
action	Action associated with the JDBC resource.	admin

For example, for a JDBC resource with the following properties:

category=connectionPool, resource=myPool, action=admin

The complete resource mapping (effective resource) would be:
/wlsspiaz/jdbc/connectionPool/myPool/admin

JMS Resource

To protect a WebLogic JMS resource, *resource_type_filter* must specify the following parameters (in the order shown):
/desType/resource/action

Field Name	Description	Field value examples
destType	Destination type of the JMS resource.	queue
resource	Name of the JMS resource.	myQueue
action	Action performed on the JMS resource.	receive

For example, for a JMS application with the following properties:

destType=queue, resource=myQueue, action=receive

The complete resource mapping (effective resource) would be:
/wlsspiaz/jms/queue/myqueue/receive

JNDI Resource

To protect a WebLogic JNDI resource, *resource_type_filter* must specify the following parameters (in the order shown):
/path/action

Parameter Name	Description	Field value examples
path	JNDI name in the JNDI path tree. Treat multiple JNDI names as separate slash(/)-delimited parameters.	pathComponent1, pathComponent2
action	Action to be performed on the JNDI resource.	modify

For example, for a JNDI application with the following properties:

path={pathComponent1,pathComponent2}, action=modify

The complete resource mapping (effective resource) would be:
/wlsspiaz/jndi/pathcomponent1/pathcomponent2/modify

Server Resource

To protect a WebLogic Server resource, *resource_type_filter* must specify the following parameters (in the order shown):
/server/action

Parameter Name	Description	Field value examples
server	Name of the server on which the action needs to be performed.	MyServer
action	Action performed on the Server.	shutdown

For example, for a server application with the following properties:

server=MyServer, action=shutdown

The complete resource mapping (effective resource) would be:
/wlsspiaz/svr/MyServer/shutdown

URL Resource

To protect a WebLogic URL resource, *resource_type_filter* must specify the following parameters (in the order shown):
application/contextPath/uri/httpMethod

Parameter Name	Description	Field value examples
application	Name of the application servicing this URL.	myApp
contextPath	Context-path of the application servicing this URL.	/mywebapp
uri	The URI requested. Treat multiple path arguments as separate slash(/)-delimited parameters.	/foo/bar.jsp

For example, for a server application with the following properties:

application=myApp, contextPath=/mywebapp, uri=/foo/bar/my.jsp

The complete resource mapping (effective resource) would be:
/wlsspiaz/url/myapp/mywebapp/foo/bar/my.jsp

Configure Rules for SiteMinder Authorization Provider

A rule identifies specific resources within a realm and whether to allow or deny access to those resources. Rules are the parts of policies that determine precisely which resources are protected, and which types of actions cause the rule to fire. A rule is required to allow resource request to be passed to protected WebLogic resources.

You configure rules for the SiteMinder Authorization Provider in the same manner as you would SiteMinder rules, defining one or more rules that identify:

- A specific resource to protect (Using SiteMinder resource mapping for WebLogic resources to map a WebLogic resource to a SiteMinder representation.)
- The Web Agent action that causes the rule to fire (Any appropriate action, such as **Post** for URL resources; the **Get** action for all other resource types).
- Whether to allow or deny access to the specified resource when the rule is fired

For example, a rule can specify that all EJB resources in a realm are protected for Get Agent actions. When a client attempts to access these resources, the rule fires and the policy containing the rule determines whether the consumer application can access the protected EJB application.

Note: For more information about creating rules, see the *SiteMinder Policy Server Configuration Guide.*

Configure Responses for SiteMinder Authentication and Authorization Providers

The SiteMinder Agent makes responses available for use in J2EE components. Responses pass user attributes, DN attributes, static text, or customized active responses from the Policy Server to the SiteMinder Agent. The SiteMinder Agent makes responses returned by the Policy Server available in the SmUser principal. The Policy Server returns two responses:

- Authentication Responses

  During authentication, these Policy Server responses are returned to the SiteMinder Authentication Provider, which is responsible for attaching the responses to the SiteMinder principal.

- Authorization Responses

  During authorization, these Policy Server responses are returned to the SiteMinder Authorization Provider, which is responsible for attaching the responses to the SmUser principal.

  Authorization Responses from the SiteMinder Policy Server are returned to the SiteMinder Authorization Provider during authorization. The SiteMinder Authorization Provider is responsible for attaching these responses to the SmUser Principal.

The SmUser principal provides access to responses using public interfaces. You can configure responses against WebLogic rules and they must be obtained programmatically using the following calls:

- getName ()

  Returns the name of a principal.

- getUserDN ()

  Returns the user DN of a principal.

- getSessionID ()

  Returns the session ID of a principal.

- getSessionSpec ()

  Returns the session spec of a principal.

- getAuthDirID ()

  Returns the Object ID of the user directory a principal was authenticated against.

- getAuthResponses ()

  Returns the responses returned by the Policy Server during authentication.

- getAzResponses ()

  Returns the responses returned by the Policy Server during authorization of a resource by a principal.

The following code snippet is an example that shows the WebLogic Server obtaining SmUser principals response attributes in a J2EE Servlet:

```
public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
{
  javax.security.auth.Subject subject =    weblogic.security.Security.getCurrentSubject ();
  java.util.Set set = subject.getPrincipals
  (com.netegrity.siteminder.weblogic.sspi.auth.SmWLSUser.class);
  java.util.Iterator i = set.iterator();
  while (i.hasNext())
  {
     SmWLSUser smUser = (SmWLSUser)i.next();
     // Get Authentication Responses
     HashMap auResponseMap = smUser.getAuthResponses();
     // Get Authorization Responses
     HashMap azResponseMap = smUser.getAzResponses();
  }
}
```

Limitations of Responses in the SmUser Principal

Because the SiteMinder Agent makes responses available in the SmUser Principal, there are limitations associated with availability of these responses, as noted in the following table. These limitations are due to the behavior of WebLogic Server Security Services.

J2EE Component Scenario	Authentication Responses	Authorization Responses
Web client accessing servlet. Responses requested within the servlet or JSP	Available*	Available

J2EE Component Scenario	Authentication Responses	Authorization Responses
Java client accessing EJB.Responses requested within the EJB on the server side	Available*	Available
Java client accessing the EJB. Responses request on the client side, that is, on the remote Java Virtual Machine.	Available*	Not Available**

*The Authentication responses are available after the authentication phase. During the validation phase, the authentication responses are not altered and, as a result, authentication responses are only set during the initial authentication.

**The Authorization responses are not available, as the SmUser principal is serialized to the client JVM during the authentication phase. The Authorization requests do not alter the principal. As a result, the SmUser principal is not reserialized to the client during authorization requests and authorization responses are not available inside a serialized SmUser principal on a remote JVM. Also, Authorization responses are only available for J2EE components that the responses bind to. For example, if a servlet accesses an EJB, the Authorization responses are only available for the servlet before accessing the EJB. Once the EJB is accessed by the servlet code, EJB responses are available in the EJB.

Configure Policies for SiteMinder Authorization Provider

Policies define how clients interact with your WebLogic resources. They bind rules, users and responses defined within a policy domain that define what happens when requests are sent to resources defined in a realm.

You configure policies to protect WebLogic resources using the SiteMinder Authorization Provider in the same way as you would policies to protect web resources. However, the following features are not supported:

- Policy expressions
- Impersonation

Note: For more information about creating policies, see the *SiteMinder Policy Server Configuration Guide*.

Chapter 8: Logging

This section contains the following topics:

Log File Summary

The SiteMinder Agent consists of multiple distributed components. Therefore, logging for the SiteMinder Agent is captured in different log files.

More Information

SiteMinder Agent Provider Log (see page 107)
SiteMinder Agent Connection Log (see page 107)
Log File Options (see page 108)

SiteMinder Agent Provider Log

The SiteMinder Agent Provider log records runtime messages for SiteMinder Agent security providers, logging events such as the SiteMinder Agent validating a user identity and authorizing a resource request.

You can configure a single log SiteMinder Agent Provider log or configure a separate log for each SiteMinder Agent Provider.

More Information

Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider (see page 113)
Configure SiteMinder Agent Log Files (see page 108)

SiteMinder Agent Connection Log

The SiteMinder Agent connection log file records SiteMinder Agent start-up messages and shows whether the SiteMinder Agent has made a successful connection with the Policy Server. You can use it to monitor the state of the connection between the SiteMinder Agent and the Policy Server.

By default, the SiteMinder Agent connection log, SMWLSASADefaultLog.log, is located in *ASA_HOME*\log

where *ASA_HOME* is the installed location of the SiteMinder Agent. For example:

- For Windows:

  C:\smwlsasa\log

- For UNIX:

  /opt/smwlsasa/log

To configure the SiteMinder Agent connection log, edit the smagent.properties file in the following location:

ASA_HOME\conf\smagent.properties

Note: The Default SiteMinder Agent log does not support dynamic log level changes.

More Information

Configure SiteMinder Agent Log Files (see page 108)

Configure SiteMinder Agent Log Files

The following sections discuss configuring SiteMinder Agent log files.

Log File Options

This section provides information about the parameters that you can use to configure the SiteMinder Agent Provider log and the SiteMinder Agent connection log.

You can set these parameters in the following locations:

- For the SiteMinder Agent Provider log:
  - The Agent Configuration Object in the Administrative UI
  - The Agent configuration file (WebAgent.conf)
- For the SiteMinder Agent connection log, set logging parameters in the *ASA_HOME*\smagent.properties file

Parameter	Description
LogAppend	Determines whether the SiteMinder Agent logs information to an existing log file instead of rewriting the entire file each time logging is invoked. **Note:** To use the LogAppend parameter, also specify the LogFile and LogFileName parameters.
LogConsole	Logs messages in a Command Prompt window.
LogFile	Determines whether messages are written to a file. If you set the Logfile parameter to yes, be sure to specify the location of the log file in the LogFileName parameter.
LogFileName	If the Logfile parameter is set to yes, the location and file name of the file where the SiteMinder Agent writes messages.
LogLevel	Determines the amount and type of information that is logged in a file or a console window. The log levels are: 0—No log messages, however, a log file is created. 1—Fatal messages 2—Error messages 3—Warning messages 4—Information messages 5—Trace messages
LogRollover	Determines whether the SiteMinder Agent starts a new log file after a specified period or when the log file reaches a certain size. If set to yes, a new log file is created after the amount of time specified in the LogRolloverTime parameter, or after the log reaches the size specified in the LogRolloverSize parameter.
LogRolloverSize	Indicates the maximum KB size of the log file before the SiteMinder Agent creates a new log file. The default is 10 MB (10240 KB). **Note:** The LogRollover parameter must be set to yes for this parameter to apply.

Parameter	Description
LogRolloverTime	Indicates when the SiteMinder Agent creates a new log file. Specify: ■ 1 to create a new log file every hour ■ 12 to create a new log file every 12 am or 12 pm ■ 24 to create a new log file every day ■ 168 to create a new log file every week ■ 720 to create a new log file every month The default value is every 12 hours. **Note:** The LogRollover parameter must be set to yes for this parameter to apply.

More Information

Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider (see page 113)
SiteMinder Agent Connection Log (see page 107)
Append Log Messages to an Existing Log File (see page 112)
Display SiteMinder Agent Log Messages in a Console (see page 111)
Record Messages in a Log File (see page 110)
Set Log Levels (see page 111)
Limit the Log File Size (see page 113)

Record Messages in a Log File

To write messages to a log file:

Set the logfile parameter to Yes and enter a file name in the logfilename parameter.

logfile="Yes"

logfilename="/opt/smwlsasa/log/*log_file_name*.log"

where *log_file_name* is the name of the log file where messages will be recorded.

The default for the SiteMinder Agent connection log file is:
logfilename="/opt/smwlsasa/log/SMWLSASADefault.log"

Note: To record SiteMinder Agent Provider log messages in a separate file for each SiteMinder Agent Provider, create an Agent configuration file for each Provider and specify a different value for the logfilename parameter in each file.

More Information

Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider (see page 113)

Display SiteMinder Agent Log Messages in a Console

The SiteMinder Agent supports the display of log messages in a console. However, the SiteMinder Agent does not open a new console screen for this display. Instead, it prints the log messages to the same console in which the WebLogic Application Server was started.

To write messages to the console, set the logconsole parameter to Yes in the Agent configuration file. For example:

```
logconsole="Yes"
```

Set Log Levels

You can configure the SiteMinder Agent to generate different levels of log messages and then display them in a file or console. Choosing a log level facilitates troubleshooting and debugging, as the log level determines the severity and extent of the logged messages. In addition, it provides control for the level of detail that the SiteMinder Agent includes in a log.

To change the log level, set the loglevel parameter to a log level described in the following table. For example:

```
loglevel="1"
```

Valid log levels are:

Log Level	Type of Messages
0	No log messages. Note however that log files are still created.
1	Fatal Messages only. For example, fatal messages are logged when the SiteMinder Agent fails to connect to a Policy Server during server startup.
2	Error Messages and Level "1" Messages. Error messages are logged when problems are encountered during SiteMinder Agent initialization or runtime that prevent the SiteMinder Agent from functioning correctly. For example, if the Validation realm is unavailable, an error message is logged.

Log Level	Type of Messages
3	Warning and Level "2" Messages. Warning messages are logged when the SiteMinder Agent encounters noncritical configuration problems that do not affect the functionality of the SiteMinder Agent. For example, if an incorrect value for pspollinterval is specified, the value is ignored and a default value is used.
4	Informational and Level "3" Messages. Informational messages are logged for SiteMinder Agent activity and flow.
5	Tracing and Level "4" Messages. Tracing messages provide tracing information for the SiteMinder Agent.
6	Extended debug logging and Level "5" messages. Reserved for future use. The SiteMinder Agent does not log any messages of this type at this time.

To avoid large log files, leave the log level at "1" so the SiteMinder Agent logs only critical errors. If you want to audit the activity of your site more carefully, change the log level to 4.

Note: If a noninteger log level is specified, the SiteMinder Agent defaults to log level "0".

Append Log Messages to an Existing Log File

To add logging information to an existing log file instead of creating a new file each time the SiteMinder Agent is initialized:

Enable the logappend parameter as follows:

logappend="Yes"

Limit the Log File Size

To prevent log files from getting too large, set the logrollover parameter to Yes and set the maximum KB size of the file in the logrolloversize parameter. After the maximum KB size is reached, a new log file is created.

```
logrollover="Yes"
logrolloversize="10240"
```

The default size is 10 MB (10240 KB).

Note: To create a new log after a specified period, instead of when it reaches a maximum size, use the logrollovertime parameter. Set the value in hours. For example, to create a new log file every 12 hours, set the logrollovertime parameter to 12.

Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider

You can configure SiteMinder to write provider-specific messages to a separate log file for each provider. For example, you can configure one log file for Identity Asserter messages and a different log file for Authentication Provider messages.

You can configure log parameters:

- Centrally, by setting the log parameters in the Agent Configuration Object. In this case, the settings apply to all of the logs.

  Note: Most settings in the Agent Configuration Object are dynamic. You only have to restart the WebLogic Server for changes to take effect if one or more changed settings are static.

- Locally, by setting the log parameters in a configuration file for each provider. In this case, the settings apply only to the provider with which the configuration file is associated.

To configure separate logs for each SiteMinder Agent provider

1. Remove the LogFileName parameter from the Agent Configuration Object configured for the SiteMinder Agent.
2. Create an Agent configuration file for each SiteMinder Agent provider as described in Create an Agent Configuration File for Each SiteMinder Agent Provider.
3. In the configuration file where the SiteMinder Agent is installed:
   a. Add the logfilename parameter and specify the location and name of the log file as follows:

      LogFileName=*path to log file*

      For example:

      For Windows: LogFileName=C:\smwlsasa\log\IA.log

      For UNIX: LogFileName=/opt/smwlsasa/log/IA.log

   b. Optionally, configure additional logging parameters.

   Note: The log parameters in the configuration file override the log parameters in the Agent Configuration Object if AllowLocalConfig parameter is set to yes.
4. Repeat this process for each provider that requires a separate log file.

More Information

Configure SiteMinder Agent Log Files (see page 108)

Configure Identity Manager SDK Logging Properties File

If the SiteManager user directory is configured in an CA Identity Manager environment and the SiteMinder Server Agent is returning physical group membership to the WebLogic Server, you can configure CA Identity Manager SDK logging. Logging is configured using the log4j.properties file, which is not installed with the SiteMinder Agent.

To get this file, copy the log4j.properties file from the CA Identity Manager installation to the ASA_HOME\conf directory. The log4j.properties file specifies the logging properties for the CA Identity Manager SDK, which is part of the CA Identity Manager imsapi6.jar file.

The following is a sample log4j.properties file for logging to a console:

```
log4j.rootCategory=DEBUG, imrexport
log4j.appender.imrexport=org.apache.log4j.ConsoleAppender
log4j.appender.imrexport.layout=org.apache.log4j.PatternLayout
log4j.appender.imrexport.threshold=INFO
```

Chapter 9: Verify the SiteMinder Agent Installation and Configuration

This section contains the following topics:

Introduction

Use the procedures in this chapter to verify that your SiteMinder Agent for Oracle WebLogic deployment is installed and configured correctly in the WebLogic Server and the SiteMinder Policy Server.

The test scenario in this chapter uses a sample security web application, located at WLS_*HOME*/weblogic*VERSION*/samples/server/examples/build/examplesWebApp, where *WLS_HOME* is the installed location and *VERSION* is the version number of the WebLogic Server.

To verify that the SiteMinder Agent is operating correctly

1. Deploy the WebLogic sample security application.
2. Set up the test scenario.
3. Verify that the SiteMinder agent providers start correctly.
4. Access the Security web application resource in a web browser.
5. Check the SiteMinder agent provider logs.

More Information

Set Up the Test Scenario (see page 116)
Verify the SiteMinder Agent Installation and Configuration (see page 115)
Check the SiteMinder Agent Provider Logs (see page 122)

Deploy the WebLogic Sample Security Application

Add the following security constraint to the Web application deployment descriptor:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>JWS_WebService_jsp</web-resource-name>
    <url-pattern>/JWS_WebService.jsp</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint>
    <role-name>PrivilegedUser</role-name>
  </auth-constraint>
</security-constraint>
```

Deploy the sample security Web application and verify that it works properly with the authentication provider and other security settings you configured.

To verify that the sample is installed and protected correctly, use a web browser to access the following security web application resource:

http://*fully_qualified_domain_name*:7001/examplesWebApp/JWS_WebService.jsp

where *fully_qualified_domain_name* is the name of the system where WebLogic is installed. For example:

http://server1.acmewidget.com:7001/examplesWebApp/JWS_WebService.jsp

When you access this URL, WebLogic prompts you for credentials using a default realm. After the authentication and authorization process, you will be granted access to the target resource.

More Information

Set Up the Test Scenario (see page 116)

Set Up the Test Scenario

The following sections contain information about setting up the test scenario.

Modify the Sample Security Web Application

To leverage an Identity Asserter, WebLogic requires that the target web application is configured to use CLIENT-CERT authentication method.

To configure the target web application to use the CLIENT-CERT authentication method

1. Modify the web application deployment descriptor as follows:

   ```
   <auth-method>CLIENT-CERT</auth-method>
   ```

2. Redeploy the web application onto WebLogic Server.

Configure the Policy Server for the Security Application

Configure realms, rules, and a policy to protect the WebLogic Sample Security Application.

Note: The following procedure provides an overview of the steps required to create the required policy objects with appropriate parameter settings. For detailed procedural information, see the *Policy Server Configuration Guide*.

To configure the Policy Server to protect the Security Application

1. Start the SiteMinder Administrative UI.
2. Verify that the domain that contains the realms that you created for the SiteMinder Agent Providers are associated with the user directory that contains the user that you want to use for the test.
3. Create a rule for the Authorization Provider Realm with the following properties:

   Domain

   The domain that contains the realms that you created for the SiteMinder Agent Providers.

   Realm

   The realm that you created for the Authorization Provider.

   Name

   Authorization Provider Rule Test Rule.

   Resource

   *

   Action

   Select the Web Agent Actions option button and highlight the GET action.

4. Modify the realm that you created for the Authorization Provider to create a subrealm with the following properties to Protect URL Resources:

   Name

   A unique name for the subrealm.

   Resource Filter

   The resource filter URL.

   Default Resource Protection

   Protected.

   Authentication Scheme

   Basic.

5. Create a rule in the subrealm you created in Step 4 with the following properties

   Name

   A unique name for the rule.

   Resource

   /security/examplesWebApp/*

   Action

   Verify that the Web Agent actions option button is selected, and the GET action is highlighted.

6. Create a Policy with the following properties to Protect the Security Sample:

   Name

   A unique name for the policy.

   Users

   Add users or groups of users that are allowed to access the security application.

   Rules

   select the rules that you created in steps 3 and 5.

Enable the SiteMinder Agent Providers

If the SiteMinder Identity Asserter, Authentication Provider, Authorization Provider, and Adjudication Provider are not enabled, enable them using the EnableWebAgent parameter:

```
EnableWebAgent="Yes"
```

You set the EnableWebAgent parameter in the Agent configuration file in *ASA_HOME*\conf.

Depending on your configuration, you may have a separate Agent configuration file for each SiteMinder Agent Provider.

More Information

Create an Agent Configuration File for Each SiteMinder Agent Provider (see page 52)

Configure Logging

Configure the SiteMinder Agent Provider and Connection logs before running the verification test. If the test fails you can use these logs to confirm that the verification test is successful or to help you troubleshoot problems.

More Information

Logging (see page 107)

Configure the SiteMinder Adjudication Provider

To help ensure that the SiteMinder Authorization Provider authorizes the user during the verification test, configure the SiteMinder Adjudication Provider to place a higher precedence on SiteMinder authorization decisions than on decisions from other providers.

To configure the SiteMinder Adjudication Provider

1. If necessary, start the WebLogic server and the WebLogic Server Administration Console.
2. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.
3. Click on the name of the realm you are configuring (for example, myrealm).
4. Click the Providers tab.
5. Click the Adjudication tab to display the Adjudication Providers list.

6. Click the entry for your SiteMinder Adjudication Provider to open it for editing.
7. Click the Provider Specific tab.
8. Verify that the SiteMinder Permission Decision field is set to SITEMINDER_PRECEDENCE.
9. Click Save.

Verify that the SiteMinder Agent Providers Start Correctly

1. Start the Policy Server and then the WebLogic Server if they are not already running.
2. Check the WebLogic console window for any warnings or errors.
3. In the WebLogic Server Administration Console:

   a. In the navigation frame on the left of the console, click the Security Realms node in the Domain Structure list.

   b. Click on the name of the realm you are configuring. (for example, myrealm).

   c. Click the Providers tab.

   d. Verify that the Providers you created appear under the appropriate tab. For example, verify that the SiteMinder Authentication Provider appears under the Authentication tab.

4. Check the SiteMinder Agent Provider and Connection logs.

   If the Providers start successfully, you should see the following messages:

   - In the SiteMinder Agent Connection Log:

     ```
         The SiteMinder Agent is initializing.
         AdministrationManager is trying to create configuration for the SiteMinder Agent
     Creating agent connection using file Path_to_Agent_Configuration_File
     Registering the Configuration Manager with the Policy Server
     ```

   - In the SiteMinder Agent Provider log or logs:

     ```
     The SiteMinder Provider_type Provider has been successfully initialized.
     ```

If the SiteMinder Agent Providers do not start correctly, troubleshoot the configuration.

More Information

Troubleshoot the SiteMinder Agent (see page 135)

Access the Security Web Application Resource in a Web Browser

To access the security web application resource after setting up the test environment

1. Verify that the Policy Server and WebLogic Server are running.
2. In a browser, access the security web application resource through the web server at the following URL:

   http://*fully_qualified_domain_name*:*port*/examplesWebApp/JWS_WebService.jsp

   fully_qualified_domain_name

   Specifies the name of the computer where SiteMinder Agent is installed.

   port

   Specifies the port number.

   For example:

   http://server2.mycompany.com:7001/examplesWebApp/JWS_WebService.jsp

The WebLogic Server should challenge you for credentials using the Basic authentication scheme. After providing credentials, you should see a page that enables you to change the background color of the application.

To confirm that everything is working as expected, check the SiteMinder Agent Provider and WebLogic log files.

More Information

Logging (see page 107)

Check the SiteMinder Agent Provider Logs

If the SiteMinder Agent Providers are configured correctly, you should see the following messages in the SiteMinder Agent Provider logs:

- For the SiteMinder Identity Asserter:

  ```
  The SiteMinder Identity Asserter has been successfully initialized.
  ```

The SiteMinder Identity Asserter is propagating the user identity:
ID to the WebLogic server.

- For the SiteMinder Authentication Provider:

  ```
  The SiteMinder Authentication Manager received login request for user USER_DN.
  Password not shown.
  The login request succeeded.
  ```

- For the SiteMinder Authorization Provider:

  ```
  The SiteMinder Resource Manager is checking if the resource
  /wlsspiaz/security/security/admin/edit.jsp is protected
  The SiteMinder Authorization Manager is checking if user : User DN can perform actions against resource
  /wlsspiaz/security/security/admin/edit.jsp
  User : User DN is AUTHORIZED to action "GET" on resource
  /wlsspiaz/security/security/admin/edit.jsp
  ```

  Note: Verify that the resource path and the user ID are correct.

- For the SiteMinder Adjudication Provider:

  ```
  The SiteMinder Resource Manager is checking if the resource
  The SiteMinder Adjudicator is using Adjudication Policy: POLICY_NAME
  The SiteMinder Adjudicator AUTHORIZES access to resource.
  ```

Appendix A: SiteMinder Agent Installation and Configuration Files

This section contains the following topics:

SiteMinder Agent Directory Structure

The following table lists the directory structure of the SiteMinder Agent installation program.

Install location	Files included	Description
smwlsasa/bin	**Windows** smreghost.bat **UNIX** smreghost.sh	SiteMinder tool to register a trusted host.
smwlsasa/conf	WebAgent.conf smagent.properties SmHost.conf	SiteMinder Agent configuration files.
smwlsasa/log	SMWLSASADefaultLog.log	SiteMinder Agent log directory
smwlsasa/lib	sm_cryptoj.jar	Container for SiteMinder Agent Java Cryptography Extension (JCE) class files.
	smagentapi.jar	Container for all class files in the Java Agent API.
	smclientclasses.jar	Container for required class files for Java-clients that request EJBs protected by

Install location	Files included	Description
		SiteMinder
	smjavasdk2.jar	Container for all class files in the DMS API.
smlwasasa/asa-wls -uninstall	uninstaller.jar	Uninstalls the SiteMinder Agent.

In addition, the smsecurityproviders.jar file is placed in the *WLS_HOME*/server/lib/mbeantypes directory and log4j.jar is placed in the *WLS_HOME*/server/lib directory.

Modify Configuration Files

To customize the SiteMinder Agent configuration, you can modify the following settings:

- Agent configuration settings
- Trusted Host configuration settings

Guidelines for Modifying Configuration Settings

- Do not add extra spaces between these elements of the parameter settings:
  - Parameter name
  - An equal sign (=)
  - Attribute value
- Always enter quotation marks around the parameter value.
- Restart the WebLogic server after you have updated and saved configuration files, such as WebAgent.conf and SmHost.conf.

Agent Configuration Parameters

Agent configuration settings are defined in two locations:

- **Agent Configuration Object**—If you are using central agent configuration, holds parameters for the SiteMinder Agent . Agent configuration always begins with the creation of the Agent Configuration Object, which you create using the Administrative UI.

  Note: The SiteMinder Agent for Oracle WebLogic does not use the same agent configuration parameters as a SiteMinder Web Agent and even where parameters have similar names their values may not be compatible. Do not attempt to use the Agent Configuration Object for a SiteMinder Web Agent for the SiteMinder Agent for WebLogic.

- **Agent Configuration file** (WebAgent.conf)—Holds parameters for each Web Agent in a text file. You can create a separate Agent Configuration file for each SiteMinder Agent security provider to set different parameters for each provider. See Set Up the Agent Configuration File (WebAgent.conf).

  The default WebAgent.conf file is located in the following directory:

  Windows: C:\smwlsasa\conf

  UNIX: /opt/smwlsasa/conf

The following table describes the configuration parameters that apply to the SiteMinder Agent. With few exceptions (noted in this table), each parameter is valid for the Agent Configuration Object or the Agent configuration file. If you modify nondynamic parameters, then you must restart the WebLogic Server for the changes to take effect; modifying dynamic parameters does not require a restart.

Parameter Name	Value	Description
AcceptTPCookie (Dynamic; Identity Asserter only)	yes or no	Single sign-on requires a single sign-on cookie. By default, this cookie is created and written to the user's browser by SiteMinder or by a custom agent. To enable support for SDK third-party cookies, set the AcceptTPCookie to yes.
AgentConfigObject (Not dynamic) (Applies only in Agent configuration file)	String	The name of the Agent configuration object.

Parameter Name	Value	Description
AllowLocalConfig (Not dynamic) (Applies only in Agent Configuration Object)	yes or no	If AllowLocalConfiguration is set to yes, parameters set locally in the Agent configuration file take precedence over parameter settings in the Agent Configuration Object. If you want to configure a separate log for each provider, set AllowLocalConfiguration to yes. See Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider for more information.
AuthCacheSize (Dynamic)	Number	Maximum size of the Authentication cache. When the maximum size is reached, new entries replace the least recently used entries. The default value is 0.
AzCacheSize (Dynamic; Authorization Provider only)	Number	Maximum size of the Authorization cache. When the maximum size is reached, new entries replace the least recently used entries. The default value is 0.
CacheTimeout (Dynamic)	Number	The life time (in seconds) of cache entries. **Note:** This setting applies to all caches.
ChallengeForCredentials (Dynamic; Identity Asserter only)	yes or no	Specifies whether the SiteMinder Identity Asserter challenges for credentials. Default is NO.
DefaultAgentName (Dynamic)	String	The name of the Agent identity that you created in Configure the SiteMinder Policy Server for the SiteMinder Agent Providers.
EnableWebAgent (Not Dynamic) (Applies only in Agent configuration file)	yes or no	■ Enables the SiteMinder Agent. If you create a separate Agent configuration file for each provider, enable the Provider in each configuration file.

Parameter Name	Value	Description
EncryptAgentName (Dynamic; Identity Asserter only)	yes or no	Specifies whether the agent name is encrypted when redirecting to the SiteMinder Web Agent for SiteMinder IA credential collection. *Must* match the value of the same parameter on the Web Agent responsible for advanced authentication. Default is NO.
FccCompatMode (Dynamic; Identity Asserter only)	no	Specifies whether to handle backward compatibility of forms credential collection, which the SiteMinder IA does not support. Therefore set this parameter to NO for *both* the SiteMinder IA *and* the Web Agent responsible for advanced authentication. For example: `fcccompatmode="NO"`
filterdomainname (Dynamic)	yes or no	To have the SiteMinder Agent remove the domain name from the user ID string before asserting an identity, set the filterdomainname parameter to yes. Setting the value to yes allows the SiteMinder Agent to use an NTLM authentication scheme because the user identity passed from a Web Agent on a front-end proxy server to the SiteMinder Identity Asserter contains the domain name when using this authentication scheme. The default value is no.
HostConfigFile (Not Dynamic) (Applies only in Agent configuration file)	String	The name of the Host Configuration Object that you created in Configure the SiteMinder Policy Server for the SiteMinder Agent Providers.
IgnoreExt (Dynamic; Authorization Provider only)	Comma-separated string	Specifies common file extensions (.gif, .jpg, .jpeg, .png, and .class) that the Authorization Provider can ignore. The Authorization Provider passes requests for files with these extensions directly to WebLogic without authorization. Use this parameter to specify extensions of files that do not require as much security as other resources.

Parameter Name	Value	Description
IgnoreQueryData (Dynamic; Identity Asserter only)	yes or no	Specifies whether the SiteMinder Agent will cache the entire URL (including the query strings) and send the entire URI to the Policy Server for rule processing. *Must* match the value of the same parameter on the Web Agent responsible for advanced authentication. Default is NO.
LegacyEncoding (Dynamic; Identity Asserter only)	no	Specifies whether to replace any dollar sign ($) characters in legacy URLs with a hyphen (-), which the SiteMinder IA does not support. Therefore set this parameter to NO for *both* the SiteMinder IA *and* the Web Agent responsible for advanced authentication. For example: legacyencoding="NO"
LogAppend (Dynamic)	yes or no	To add logging information to an existing log file instead of rewriting the entire file each time logging is invoked, set the LogAppend parameter to yes. The default LogAppend value is no. **Note:** To use the LogAppend parameter, also specify the LogFile and LogFileName parameters.
LogConsole (Dynamic)	yes or no	To display log messages in a Command Prompt window, set the LogConsole parameter to yes. The default LogConsole value is no. Before you enable this option on an iPlanet Web Server, change the iPlanet service to interact with the desktop.
LogFile (Dynamic)	yes or no	Determines whether messages are written to a file. The default LogFile value is no. If you set the Logfile parameter to yes, be sure to specify the location of the log file in the LogFileName parameter.
LogFileName (Dynamic)	String	Location and file name of the file where the SiteMinder Agent writes messages if the Logfile parameter is set to yes.

Parameter Name	Value	Description
LogLevel (Dynamic)	Numbers 0-5	Determines the amount and type of information that is logged in a file or a console window. The log levels are: 0—No log messages, however, a log file is created. 1—Fatal messages 2—Error messages 3—Warning messages 4—Information messages 5—Trace messages The default log level is 0.
LogRollover (Dynamic)	yes or no	Determines whether the SiteMinder Agent starts a new log file after a specified period or when the log file reaches a certain size. If set to yes, a new log file is created after the amount of time specified in the LogRolloverTime parameter, or after the log reaches the size specified in the LogRolloverSize parameter. The default LogRollover value is no.
LogRolloverSize (Dynamic)	Number of kilobytes (Kb)	Indicates the maximum size of the log file before the SiteMinder Agent creates a new log file. The default is 10 MB (10240 KB). **Note:** The LogRollover parameter must be set to yes for this parameter to apply.
LogRolloverTime (Dynamic)	Number of hours	Indicates when the SiteMinder Agent creates a new log file. For example, specify 1 to create a new log file every hour; specify 168 to create a new log file every week; and specify 720 to create a new log file every month. The default value is 12 hours. **Note:** The LogRollover parameter must be set to yes for this parameter to apply.

Parameter Name	Value	Description
PersistentCookies (Dynamic; Identity Asserter only)	yes or no	Specifies whether the agent allows single sign-on for multiple browser sessions. When PersistentCookies is enabled, users who authenticate during one browser session will retain single sign-on capabilities for subsequent browser sessions. Default is NO.
ResourceCacheSize (Dynamic)	Number	The maximum number of resource cache entries that the Agent tracks. When the maximum number of entries is reached, new records replace the oldest records. The default value is 1000 entries for IIS and Domino and 750 entries for Apache and iPlanet. If you set this value to a high number, be sure that sufficient server memory is available.
ServerErrorFile (Dynamic; Identity Asserter only)	String	Specifies a page to redirect a request to if a processing error is encountered. This can either be an HTTP or local file system resource. For example: servererrorfile="http://server.ca.com:88/errorpage.html" If this setting is not configured, a default message is output to the response when the IA encounters an error. The default message is "SiteMinder Agent encountered an error while handling request. Please ask the administrator to look for messages in the agent log to check for the cause."
SMUserDirectory (Not Dynamic)	String	The user directory structures used in the SiteMinder provider authentication realm.
SMAdminUserName (Not Dynamic)	String	The user name of the SiteMinder administrator created during the Policy Server installation who has full permissions to manage all SiteMinder domain objects and users.

Parameter Name	Value	Description
SMAdminUserPassword (Not Dynamic)	Encrypted string value	The encrypted password for the SiteMinder administrator. Warning! This password can only be encrypted in the Agent Configuration Object in the Policy Server User Interace and not in the WebAgent.conf file.

More Information

Set Log Levels (see page 111)
Enable and Disabe the SiteMinder Identity Asserter (see page 68)
Enable and Disable the Authentication Provider (see page 83)
Enable and Disable the Authorization Provider (see page 89)
Enable and Disable the Adjudication Provider (see page 93)

Trusted Host Configuration

The SmHost.conf file results from a successful registration of a unique host name as a trusted host. The SiteMinder Agent installation program automatically launches the smreghost registration tool, which in turn creates the SmHost.conf file and places it in the *ASA_HOME*/conf folder.

Sample SmHost.conf file:

```
hostname="dualsol184-asa60"
hostconfigobject="cadell2k24"
policyserver="138.42.223.24,44441,44442,44443"
requesttimeout="60"
sharedsecret="{RC2}3gpfTl6uQY7BrzXbQ88G3be50bR6JGYn/oXpjLrWH2sX4eRvn4aQ5987RXeis
COH2/v5bz2Q/1/k4+N2zNgysHSEdHDWjXWAReRUxPT3gUFBoOxllQ1pKdunZa/Pbm+fwIKOl83goIyLe
WGXDuSfo9EeW7Mj+GKGI6JbXlYE2PjwiDdDTjpQomxpTXwUqSFr"
sharedsecrettime="1167758161"
```

Note: For information about trusted hosts and the parameters in the file, see the *SiteMinder Web Agent Installation Guide*.

Note: To register a trusted host outside the SiteMinder Agent installation process, run smreghost through the command line.

A trusted host is a client that is registered with the Policy Server and is, therefore, allowed to connect to the Policy Server. You can modify Trusted Host configuration settings in two places:

- Host Configuration file (SmHost.conf)—holds initialization parameters for the Trusted Host. Once the Trusted Host connects to a Policy Server, the Trusted Host uses the settings in the Host Configuration Object named in the hostconfigobject parameter of SmHost.conf.
- Host Configuration Object—holds parameters for a Trusted Host. Except for initialization parameters, Trusted Host parameters are always maintained in a Host Configuration Object.

The following describe the parameters in the SmHost.conf file and the Host Configuration Object, respectively.

Note: For information about setting these parameters, see the *SiteMinder Policy Server Configuration Guide*.

Parameter	Description	Default Value
CryptoProvider	Specifies the encryption method used for hardware encryption	BSAFE
HostName	A unique name that represents the host to the Policy Server	The name you specify when you register the trusted host.
PolicyServer	The server IP address and port numbers for the Policy Server that the Trusted Host accesses	The IP address and port numbers that you specify when you register the trusted host.
RequestTimeout	Specifies the number of seconds that the Trusted Host waits before deciding that a Policy Server is unavailable.	60
SharedSecret	An automatically generated encryption key used for encrypting traffic between the trusted host and the Policy Server. **Important!** Do not change the shared secret.	N/A

Note: You cannot modify the cryptoprovider or hostname parameters directly. For information about changing these parameters, see the *SiteMinder Web Agent Installation Guide*.

Parameter	Description	Default Value
EnableFailover	Determines which operation mode the Trusted Host uses to work with the Policy Server.	N/A
HostConfigObject	The name of the Host Configuration Object specified in the Policy Server	The name you specify when you register the trusted host.
MaxSocketPerPort	Defines the maximum number of TCP/IP connections used by the Trusted Host to communicate with the Policy Server.	20
MinSocketPerPort	Determines the number of TCP/IP connections open for SiteMinder services when you start up Policy Server services.	2
NewSocketStep	Specifies the number of TCP/IP connections that the Agent opens when new connections are required.	2
PolicyServer	The server IP address and port numbers for the Policy Server that the Trusted Host accesses	The IP address and port numbers that you specify when you register the trusted host.
RequestTimeout	Specifies the number of seconds that the Trusted Host waits before deciding that a Policy Server is unavailable.	60

Appendix B: Troubleshoot the SiteMinder Agent

This section contains the following topics:

Prepare to Troubleshoot the SiteMinder Agent

The following table describes the log files that you can use to troubleshoot problems with the SiteMinder Agent, and provides references to configuration information for each log file.

Log	Description
The SiteMinder Agent Connection log	Records SiteMinder Agent start-up messages and shows whether the SiteMinder Agent has made a successful connection with the Policy Server
The SiteMinder Agent Provider log or logs	Records runtime messages for SiteMinder Agent Providers To simplify troubleshooting tasks: ■ Create separate log files for each SiteMinder Agent Provider as described in Configure a SiteMinder Agent Provider Log for Each SiteMinder Agent Provider. ■ Set the log level for each log to "5" to display the maximum amount of information in the logs.
WebLogic Server log	Records information about WebLogic events, including information about security provider activity

More Information

Logging (see page 107)
Configure the WebLogic Server Log (see page 136)

Configure the WebLogic Server Log

You can configure the SiteMinder Agent to write error messages to the WebLogic log file, *server_name*.log (for example, myserver.log for the WebLogic Default Server).

To configure WebLogic logging In the WebLogic Administration Console

1. Start the WebLogic server and the WebLogic Server Administration Console, if necessary.
2. In the navigation frame on the left of the Console, expand the Environment node and then click the Servers node in the Domain Structure list.
3. Click on the name of the server you are configuring (for example, AdminServer(admin).
4. Change the server logging severity level:
   a. Click the Logging tab.
   b. Click Advanced to expand the available options.
   c. Select INFO from the Stdout Severity Threshold drop-down list.
5. Select Debugging options:
   a. Click the Debug tab.
   b. Expand WebLogic.
   c. Expand Security.
   d. Select the rolemap, atn, and atz entries.
   e. Click Enable.
   f. Click Activate Changes.

Solve Host Registration Problems

The following table provides information about troubleshooting common SiteMinder Agent host registration problems.

Possible Cause	Solution
The JVM has not been patched for unlimited cryptography with the Java Cryptography Extension (JCE) package.	Check for messages that indicate that the host could not be registered or with cluster failed messages. Patch the JVM for unlimited cryptography with the Java Cryptography Extension (JCE) package.

Possible Cause	Solution
Host configuration object has not been configured or Policy Server is not running.	Check for any messages that indicate the host could not be registered. Add a host configuration object and run smreghost tool separately from the installation procedure. To run this tool, see the *SiteMinder Web Agent Installation Guide*.
Trusted Host Name exists in the Policy Server.	Check for any messages that indicate the host could not be registered. Configure with another trusted host name or delete the already existing one.

Solve Installation Problems

The following table provides information about troubleshooting common SiteMinder Agent installation problems.

Symptom	Possible Cause	Confirm	Resolve problem
Execute permission denied when trying to run installation program on UNIX platforms.	Incorrect command line execution.	Enter this command in directory where installation program is stored: %> ls -l Check permissions for the installation directory; owner has -rw-rw-rw	Run sh ./[install] or run chmod to add execution privileges for the user running the installation.
Message during install that the installer could not register a host	The host configuration parameters are not set correctly.	Check the settings in the Host Configuration Object in the Policy Server user interface.	Correct any configuration problems. See Trusted Host Configuration.

Symptom	Possible Cause	Confirm	Resolve problem
Message during install that WebLogic home is corrupted.	Selected Oracle home directory instead of WebLogic home directory.	The WebLogic home directory is a subdirectory under the Oracle home.	Use the GUI installation program to navigate to the correct WebLogic home directory.

Solve Runtime Problems

The following table provides information about troubleshooting common SiteMinder Agent runtime problems.

Symptom	Possible Cause	Confirm	Resolve problem
401 Not Authorized; no identity propagation.	The web application Login Config is not set to CLIENT-CERT, which prevents the WebLogic server from triggering the SiteMinder Identity Asserter.	Try to access the application through the WebLogic httpd listener (default port 7001). Are you challenged by BASIC or FORM?	Edit web.xml manually and change the <login-config><auth-method> value to CLIENT-CERT. OR Open WebLogic Builder or other deployment descriptor GUI editor and edit the web application deployment descriptor. Change the Auth Method value under Login Config.
401 Not Authorized; no identity propagation. (continued)	Require authentication by separate authentication provider.	Try authenticating from this user directory in another method; for example, add group to the WebLogic admin role and start the WebLogic server with that username.	If DefaultAuthentication Provider still exists, edit properties and verify Control Flag is not "REQUIRED".

Symptom	Possible Cause	Confirm	Resolve problem
401 Not Authorized; no identity propagation. (continued)	SiteMinder Agent Provider not enabled.	Check the SiteMinder Agent connection log file for the Provider (or console, if enabled) for this message: *"Provider* is disabled" where *Provider* is the name of the Provider.	Edit the appropriate Agent configuration file in ASA_HOME/conf and set enableWeb Agent="Yes"

Identity Asserter Runtime Problems

Symptom	Possible Cause	Confirm	Resolve problem
Application works but identity does not propagate.	Identity Asserter not invoked.	Try accessing the application through the WebLogic httpd listener (default port 7001). Can the application be accessed without challenge?	There must be a security constraint levied against the URL; check the Web deployment descriptor.
Failed to decrypt SMSESSION cookie.	Discrepancy between Policy Server used to set session cookie and that used by Identity Asserter.	Check the Agent Configuration files for the proxy server Web Agent and the SiteMinder Agent to verify that they point to policy servers that use the same key store.	Verify that policy and key stores are synchronized. See *the SiteMinder Policy Server Administration Guide* for more information about key stores.

Symptom	Possible Cause	Confirm	Resolve problem
Single sign-on between a custom agent built with SiteMinder SDK and WebLogic does not work	The SiteMinder Identity Asserter is not configured to accept cookies from custom agents, which are different from standard Web Agent cookies.	Check the accepttpcookie value in the Identity Asserter WebAgent.conf file.	The accepttpcookie parameter must be set to "Yes" to propagate identity.
Server error on first request to WebLogic.	Required version of Web Agent not available.	Refresh the request and see if the application is now displayed.	Upgrade to the latest Web Agent QMR version.
SSL error message: "The SiteMinder Identity Asserter failed to validate the X.509 certificate" in the SiteMinder Identity Asserter Validation Realm.	There is a problem with the authentication scheme for the SiteMinder Identity Asserter Validation Realm.	See the "Resolve problem" column.	Protect the SiteMinder Identity Asserter Validation Realm with a X509 Client Cert authentication scheme.
SSL error message: "keytool error:java.lang.Exception: Failed to establish chain from reply"	There is a problem with the certificate.	See the "Resolve problem" column.	Import the server certificate in "Base 64 encoded certificate with CA certificate chain in pkcs7 format".

Solve Configuration Problems

The following table provides information about troubleshooting common Identity Asserter configuration problems.

Symptom	**Possible Cause**	**Confirm**	**Resolve problem**
No option to create SiteMinder Identity Asserter, Authentication Provider, Authorization Provider, or Adjudication Provider in WebLogic console.	MBean not deployed.	Check *WLS_HOME*/weblogic*VERSION*/server/lib/mbeantypes for the siteminder smsecurityproviders.jar file	If the file is not present on server, try rerunning the SiteMinder Agent install.
The WebLogic Server Administration Console does not display the security realm and other nodes in the left navigation pane.	The user account that is accessing the WebLogic console has not been granted the WebLogic admin role, which allows users to perform WebLogic administrative functions.	Check the WebLogic console for related messages.	Verify that the admin role is returned for the admin user starting the WebLogic server.
No log file created for a SiteMinder Agent provider.	Incorrect file path.	Check that directories exist in file path for logfilename parameter in the Agent configuration file and smagent.properties files. Pathnames should be absolute; file does not have to exist previous to start.	Correct logfilename parameter.

Symptom	Possible Cause	Confirm	Resolve problem
Problem instantiating authentication provider: java.lang.NullPointerException.	SiteMinder jar file(s) or conf directory missing from classpath.	Check -classpath switch that is echoed in WebLogic server console. Four CA jar files should be included, plus the /conf directory of *ASA_HOME*.	Set the CLASSPATH environment variable, or add the jar files to the SMASA_CLASSPATH variable in the startWebLogic script(s).
java.lang. Unsatisfied LinkError: java_agent_api	Java Agent API is not included in Java Library Path.	Verift that the Java Library Path echoed in the WebLogic server console. It should include the /bin directory of *ASA_HOME*.	Set library path variable to include the /bin directory.
Could not create an Agent Config Object.	SiteMinder Agent Provider could not determine its configuration	Check the Policy Server console for messages such as "Failed to create agent configuration for...".	In the Administrative UI, verify that the following objects exist: ■ AgentConfigObject referenced in the Agent Configuration file ■ HostConfigObject referenced in the Host Configuration file

Group Membership Problems

Symptom	Possible Cause	Confirm	Resolve problem
The SiteMinder Agent displays the message: "SmUser Authentication Directory ID not found. No Groups will be returned".	The user directory that IMS or DMS is configured for is missing in the SiteMinder Agent domain.	In the Administrative UI, verify that the domain of the SiteMinder Agent contains the user directory that IMS or DMS is configured for in the Agent Configuration Object.	If the user directory is not in the Policy Server, add it to the SiteMinder Agent domain using the Administrative UI and restart the WebLogic Server.
Verify that you correctly followed the steps in SiteMinder User Directory Not Configured in CA Identity Manager Environment (Use DMS API).	1. Verify that smjavasdk2.jar is present in the: ■ *ASA_HOME*/lib directory ■ classpath. 2. In the Policy Server User Interface, verify the: ■ value for the SmUserDirectory parameter in the Agent Configuration Object is correct. ■ SiteMinder user directory has the appropriate administrator credentials in the "Credentials and Connection" tab of the SiteMinder User Directory dialog.		

WebLogic Server Startup Problems

Symptom	**Possible Cause**	**Confirm**	**Resolve problem**
The WebLogic Server fails to start.	The CA classes are not available to the server	Check the WebLogic console or log for the following error: com.netegrity.siteminder.agentcommon.utils.g:Could not parse Input Stream	Check the CLASSPATH environment variable in the startWebLogic file, or in the shell environment
	Java Agent API is not included in Java Library Path.	Check the Java Library Path echoed in the WebLogic server console. It should include the /bin directory of *ASA_HOME*.	Set library path variable to include the /bin directory.
The WebLogic Server fails to start. (continued)	WebLogic cannot find the Agent configuration file (WebAgent.conf) for the SiteMinder Agent	Check the WebLogic console or log for the following error: Incorrect path to file *Path_to_WebAgent.conf* SiteMinder Authentication Provider Failure. Incorrect path to file	verify the Agent configuration file exists in the path specified in the error message. If using relative paths, check the following: ■ The smasa.home environment parameter is set correctly in the startWebLogic file, or ■ the Agent configuration file is located in *WEBLOGIC_HOME*/user_projects/domains/*YOUR_DOMAIN*

Symptom	Possible Cause	Confirm	Resolve problem
The WebLogic Server fails to start. (continued)	WebLogic cannot find the Agent configuration file for the Adjudication Provider If you do not access the Details tab when you create the Adjudication Provider in the WebLogic console, the location of the Agent configuration file is not set in the config.xml file	In the config.xml file, check the entry for the SiteMinder Adjudication Provider for the following parameters: <ext:site-minder-permission-*decision>p\Precedence_Setting*</ext:site-minder-permission-decision> <ext:sm-adjudication-provider-config-*file>Path_to_Agent_Config file*</ext:sm-adjudication-provider-config-file> If these parameters do not exist, the Adjudication Provider was not configured correctly.	Add the following parameters in the com.netegrity.weblogic.sspi.adjudicator.SiteMinderAdjudicationProvider element in the config.xml: <ext:site-minder-permission-*decision>p\Precedence_Setting*</ext:site-minder-permission-decision> <ext:sm-adjudication-provider-config-*file>Path_to_Agent_Config file*</ext:sm-adjudication-provider-config-file>
	The SiteMinder Authorization Provider is denying authorization for starting the WebLogic Server	In the SiteMinder Authorization Provider log, check for a message that resembles the following: "User: *USER_DN* is NOT AUTHORIZED for action "GET" on resource "wlsspiaz/svr/myserver/boot"	In the Administrative UI, create a policy that allows the WebLogic admin account to start the WebLogic server.

Symptom	Possible Cause	Confirm	Resolve problem
WebLogic Server failed to start. Message in the WebLogic console says that "user weblogic failed to boot the server"	You have configured the SiteMinder Authorization Provider incorrectly.	Follow the steps in the "Resolve problem" column.	If you have already configured the SiteMinder Authorization Provider: 1. Create a **/*** rule in the Authorization Provider realm. 2. Add this rule to the policy that contains all users of the user directory. If you have not configured the SiteMinder Authorization Provider, verify that the group of user "weblogic" is being returned as "Administrators". You can configure groups to be returned by using SiteMinder responses.

Symptom	Possible Cause	Confirm	Resolve problem
WebLogic Server prompts for credentials again in error. After you configure the SiteMinder Authentication, Authorization, and Adjudication providers, the WebLogic Server starts up successfully and the WebLogic console is accessible. However, any changes you make to these provider components (that is, updating a component and clicking **Apply**) in the WebLogic console are not successful and the WebLogic Server prompts you for user credentials again. Also, the WebLogic Server denies permission for any updates you perform using the WebLogic console.	You did not select the **Post** action in the Authorization Provider realm's rule in the SiteMinder Agent domain. The WebLogic Server's log files say: "..post action is not authorized..."	That you configured the Authorization Provider realm's rule correctly in the SiteMinder Agent domain.	Using the Administrative UI: 1. Check Authorization Provider realm's rule in the SiteMinder Agent domain. 2. Make sure a **/*** rule is present with **Get** and **Post** actions selected.