CA SiteMinder®

Agent for JBoss Guide

12.52

CA Technologies Product References

This document references the following CA Technologies products:

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

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Set the JBoss 5.x Environment on UNIX (see page 62)—Corrected the separator character for JBOSS_CLASSPATH entries to a colon (:). Resolves CQ 165866 and STAR issue 21264939-01. Also corrected variables to use dollar ($) characters. Resolves CQ 169680 and STAR issue 20756839-1.

Contents

Chapter 1: Overview 9

Chapter 2: Install the SiteMinder Agent 17

Chapter 1: Overview

This section contains the following topics:

Introduction

This chapter introduces the SiteMinder Agent for JBoss and describes how it integrates with the JBoss Application Server to secure J2EE resources deployed on that operating environment.

The SiteMinder Agent for JBoss provides the following two JBossSX custom security interceptors that allow it to be configured into SiteMinder and CA SiteMinder® Web Services Security environments as required:

SiteMinder Agent Security Interceptor

> The SiteMinder Agent Security Interceptor provides a SiteMinder Agent solution that provides SiteMinder access control for web application resources (including servlets, HTML pages, JSP, and image files).

WSS Agent Security Interceptor

> The WSS Agent Security Interceptor provides a SiteMinder Web Services Security (WSS) Agent solution that provides CA SiteMinder® Web Services Security access control for JAX-WS and JAX-RPC web service resources.

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 Java Authentication and Authorization Server (JAAS) and the JBossSX security framework.
- Knowledge of how to provide security constraints for J2EE components through security realms and deployment descriptors.
- Experience with configuring and managing the JBoss Application Server.
- If configuring protection for web applications, familiarity with SiteMinder concepts and terms.

- If configuring protection for web services, understanding of JAX-RPC and JAX-WS web service implementations and handlers and familiarity with CA SiteMinder® Web Services Security concepts and terms.
- Knowledge of with Policy Server configuration tasks.

SiteMinder Agent Security Interceptor

The SiteMinder Agent Security Interceptor provides an *identity assertion* solution for securing JBoss web container resources by perimeter authentication.

In the perimeter authentication model, user identity is validated outside the JBoss security domain and passed to the JBoss Application Server in the form of a token associated with the user request. An Identity Asserter configured within the JBoss security domain then obtains authenticated user information from the token.

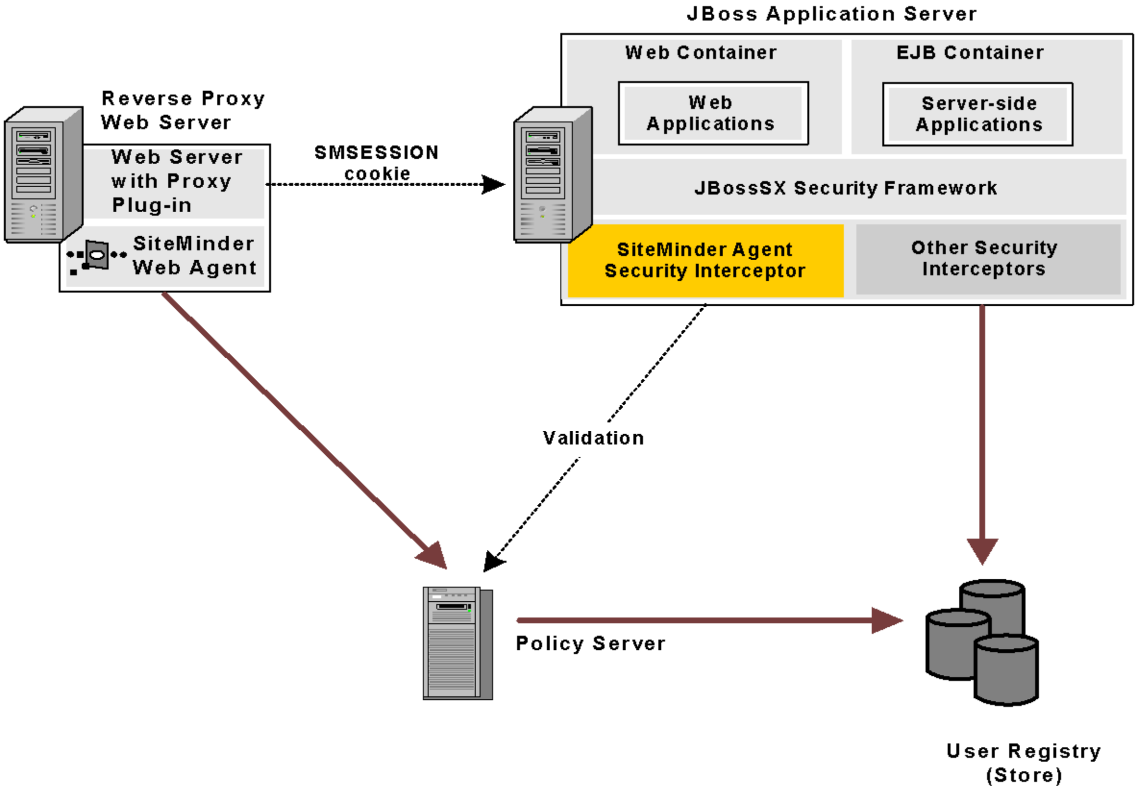
How the SiteMinder Agent Security Interceptor Works

The SiteMinder Agent Security Interceptor allows the JBoss Application Server to trust requests with associated SiteMinder session (SMSESSION) cookies so that these users are not challenged for credentials.

SiteMinder session cookies are obtained from a SiteMinder Web Agent on a proxy server configured to:

- Intercept HTTP requests for JBoss resources
- Authenticate and authorize users through policies defined on the Policy Server

- Forward requests together with user credentials (in a session cookie) to the application server as shown in the following illustration:



When you configure the SiteMinder Agent Security Interceptor as an identity asserter in a security realm, the JBossSX security framework passes any SiteMinder session cookies associated with a request for a resource within that realm to the SiteMinder Agent Security Interceptor for validation. The SiteMinder Agent Security Interceptor then:

1. Validates the token by calling the Policy Server to verify that its session is valid (SiteMinder session cookie).
2. Obtains the requester userDN from the token and maps it to a username.
3. Passes the associated username and SiteMinder session information back to the JBossSX security framework.

Note: If you must only allow access to web applications for clients with *existing* SiteMinder Single Sign-On (SSO) sessions, you can use the SiteMinder Agent Security Interceptor as a standalone component without the proxy server-related components.

SiteMinder Agent Security Interceptor Components

The SiteMinder Agent Security Interceptor consists of the following modules that you can configure into the JBossSX security framework:

- SiteMinder Agent Authenticators (see page 12)
- SiteMinder Agent Login Module (see page 13)

SiteMinder Agent Authenticators

In the JBossSX security framework, requests for web application resources in the web container are handled by default authenticators for Basic, Client-Cert, Form, and Digest authentication.

The SiteMinder Agent Security Interceptor provides the following custom replacement SiteMinder Agent Authenticators that extend the functionality of the JBoss default authenticators with the ability to authenticate a user request based on an associated SiteMinder session cookie:

SMJBossIdentityAsserter

(New) Authenticates user identity using the SiteMinder session cookie only. If there is no valid SiteMinder session cookie, the authenticator returns an authentication failure result.

SMJBossBasicAuthenticator

(Replaces JBoss default BasicAuthenticator) First attempts to authenticate user identity using the SiteMinder session cookie. If there is no valid SiteMinder session cookie, performs Basic authentication.

SMJBossFormAuthenticator

(Replaces JBoss default FormAuthenticator) First attempts to authenticate user identity using the SiteMinder session cookie. If there is no valid SiteMinder session cookie, performs Form authentication.

SMJBossClientCertAuthenticator

(Replaces JBoss default ClientCertAuthenticator) First attempts to authenticate user identity using the SiteMinder session cookie. If there is no valid SiteMinder session cookie, performs Client-Cert authentication.

SMJBossDigestAuthenticator

(Replaces JBoss default DigestAuthenticator) First attempts to authenticate user identity using the SiteMinder session cookie. If there is no valid SiteMinder session cookie, performs Digest authentication.

The SiteMinder Agent Authenticators first attempt to retrieve a SiteMinder session cookie from a request. If there is a valid SiteMinder session cookie, the SiteMinder Agent Login Module is used to authenticate the user and create user principles. If there is no valid SiteMinder session cookie, the appropriate JBossSX default authenticator functionality occurs.

SiteMinder Agent Login Module

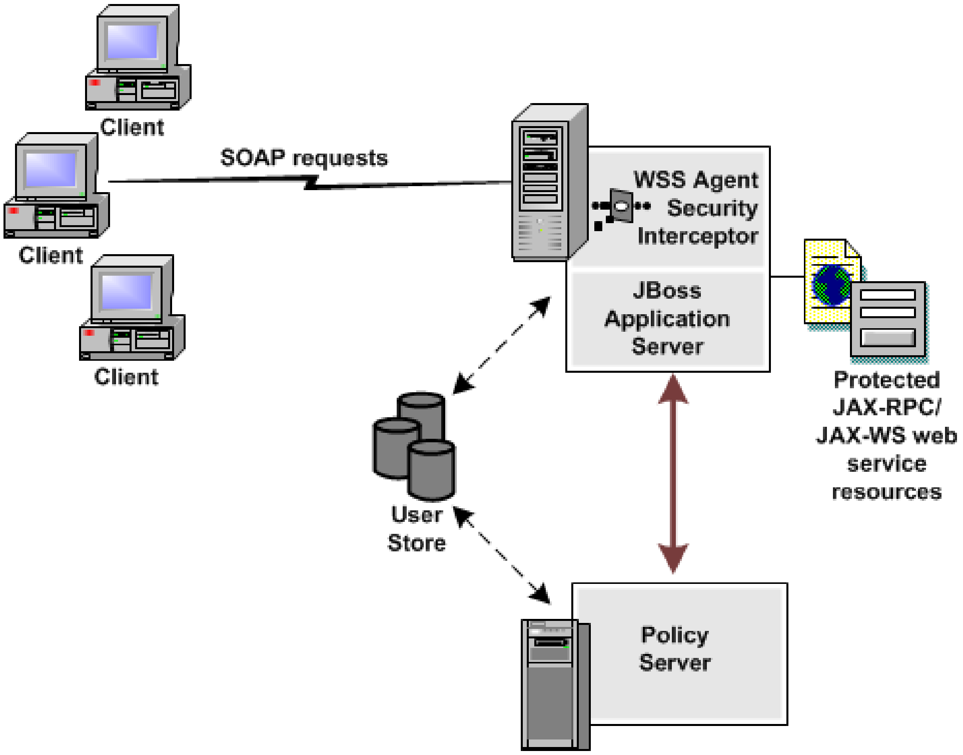
The SiteMinder Agent Login Module authenticates credentials (username/password) obtained from valid SiteMinder session cookies by SiteMinder Agent authenticators.

If SiteMinder authentication is successful, the SiteMinder Agent Login Module populates a JAAS Subject with a SiteMinder Principal that contains the username and associated SiteMinder session data.

WSS Agent Security Interceptor

The WSS Agent Security Interceptor provides a SiteMinder WSS Agent solution for the JBoss Application Server. The WSS Agent Security Interceptor integrates the JBoss Application Server into the CA SiteMinder® Web Services Security environment, enabling you to implement policy-based fine-grained access control to protect JBoss-hosted JAX-RPC and JAX-WS web service resources. The WSS Agent Security Interceptor also supports bi-directional CA SiteMinder® Web Services Security/SiteMinder and JBoss single sign-on (SSO).

A high-level overview of the WSS Agent Security Interceptor architecture is shown in the following illustration



How the WSS Agent Security Interceptor Works

When fully configured into the JBossSX security infrastructure, the WSS Agent Security Interceptor does the following:

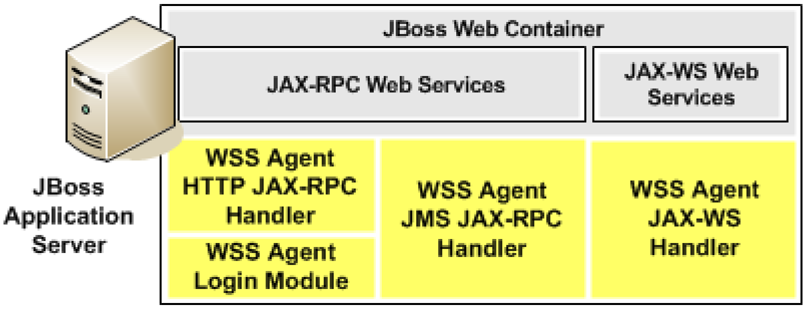
1. Intercepts SOAP requests sent over HTTP(S) or JMS transports to JAX-RPC and JAX-WS web services deployed on the JBoss Application Server.
2. Communicates with the Policy Server to authenticate and authorize the message sender
3. Upon successful authentication and authorization, passes the request message on to the addressed web service.

WSS Agent Security Interceptor Components

The WSS Agent Security Interceptor consists of the following modules that you can configure into the JBossSX security framework:

- WSS Agent JAX-WS Handler
- WSS Agent JMS JAX-RPC Handler
- WSS Agent HTTP JAX-RPC Handler
- WSS Agent Login Module

Note: You do not need to configure all WSS Agent modules, only the ones you require. WSS Agent modules can be configured globally for all web services of each type or for each individual web service.



WSS Agent JAX-WS Handler

The WSS Agent JAX-WS Handler is a custom JAX-WS Handler that intercepts requests for JAX-WS web services and authenticates credentials obtained from intercepted requests against associated user directories configured in CA SiteMinder® Web Services Security:

Note: The WSS Agent JAX-WS Handler can obtain credentials from SOAP requests or from associated SiteMinder session cookies of users with pre-established CA SiteMinder® Web Services Security and SiteMinder sessions.

If CA SiteMinder® Web Services Security authentication is successful, the WSS Agent JAX-WS Handler determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated CA SiteMinder® Web Services Security authorization policies.

WSS Agent JMS JAX-RPC Handler

The WSS Agent JMS JAX-RPC Handler is a custom JAX-RPC Handler that intercepts requests for JAX-RPC web services sent over JMS transport and authenticates credentials obtained from those requests against user directories configured in CA SiteMinder® Web Services Security.

If CA SiteMinder® Web Services Security authentication is successful, the WSS Agent JMS JAX-RPC Handler determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated CA SiteMinder® Web Services Security authorization policies.

WSS Agent HTTP JAX-RPC Handler

The WSS Agent HTTP JAX-RPC Handler is a custom JAX-RPC Handler that intercepts SOAP message requests sent to JAX-RPC web services over HTTP transport and diverts them to the WSS Agent Login Module for authentication and authorization decisions.

Note: If you configure the WSS Agent JAX-RPC Handler, you must also configure the WSS Agent Login Module.

WSS Agent Login Module

The WSS Agent Login Module is a JAAS Login Module that performs authentication and authorization for JAX-RPC web services protected by the WSS Agent HTTP JAX-RPC Handler. (Login Module functionality is built into the WSS Agent WS and JMS JAX-RPC Handlers.)

The WSS Agent Login Module can authenticate and authorize credentials obtained by the WSS Agent JAX-RPC Handler from SOAP requests or from associated SiteMinder session cookies of user with pre-established CA SiteMinder® Web Services Security and SiteMinder sessions.

If CA SiteMinder® Web Services Security authentication is successful, the WSS Agent Login Module determines whether an authenticated user is allowed to access a protected JBoss resource, based on associated CA SiteMinder® Web Services Security authorization policies.

Note: If you configure the WSS Agent Login Module, you must also configure the WSS Agent JAX-RPC Handler.

Chapter 2: Install the SiteMinder Agent

This section contains the following topics:

Installation Overview

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

- Web Application Security Interceptor (SiteMinder functionality)
- Web Service Interceptor (SOA Security Manager functionality)

Note: All components of both interceptors are installed when you run the SiteMinder Agent installation. However, you need only configure the interceptor modules that you want to use.

Install Preparation

Before you install a SiteMinder Agent for JBoss, there are a number of pieces of information you will need and requirements that must be met.

Locate the Platform Support Matrix

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

Follow these steps:

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

   The CA SiteMinder® product page appears.
4. Click Product Status, CA SiteMinder® Family of Products Platform Support Matrices.
5. Locate the **SiteMinder Agent for Application Servers 12.52** entry and open open the associated PDF file.

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

Software Requirements

Supported versions of the following software are always required before you install the SiteMinder Agent.

- JBoss Enterprise Application Platform. For hardware and software requirements, see the JBoss Enterprise Application Platform documentation.
- *One* of the following Policy Servers:
  - SiteMinder Policy Server (for web application protection)
  - Policy Server (for web service and, if also licensed for SiteMinder, web application protection)
- Java virtual machine (JVM) with the path to the JVM present in the host environment. For example, on UNIX systems, if the JVM is not in the PATH variable, run the following commands:

  ```
  PATH=$PATH:JVM/bin
  export PATH
  ```

  JVM

  Specifies the location of your Java virtual machine (for example /opt/jre1.5.0_06/bin).

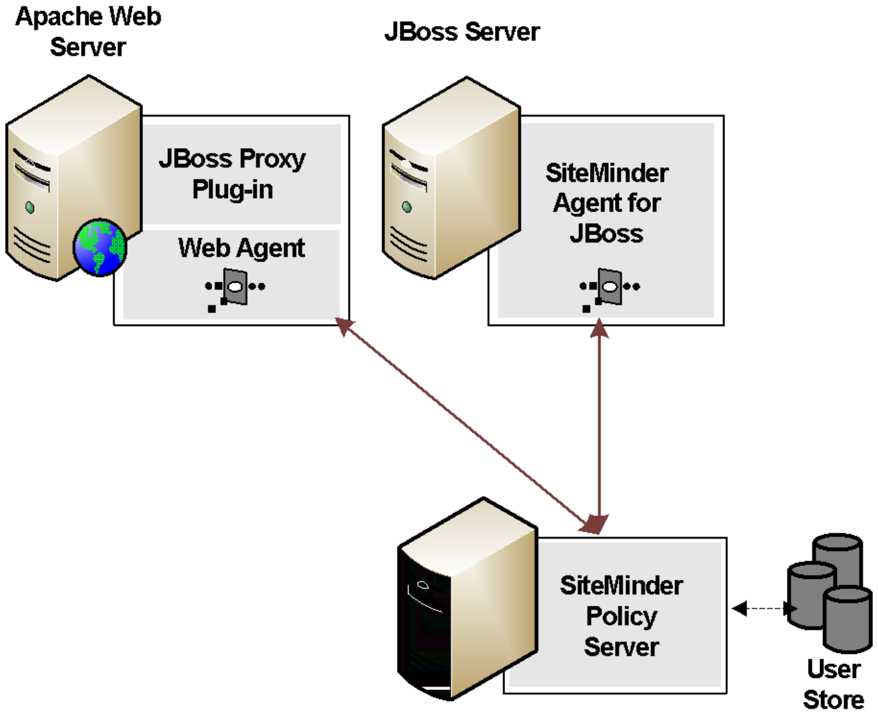
To use the SiteMinder Agent Web Interceptor to validate identities obtained from SiteMinder session cookies during perimeter authentication, the following software is also required:

- SiteMinder Web Agent
- A web server and proxy plug-in supported by SiteMinder and JBoss

  For supported web servers and proxy plug-ins, see:

  - Platform Support Matrices on the Technical Support site.
  - Supported Configurations for JBoss Enterprise Application Platform in the JBoss Enterprise Application Platform documentation.

The following illustration shows where each of these software components is installed in an environment that uses SiteMinder SSO-based perimeter authentication.



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

Installation Checklist

To install the CA SiteMinder® Agent for JBoss, complete all the steps in the following table. To help ensure proper configuration, follow the steps in order.

Complete?	Steps	For information, see...
	Install and configure a Policy Server	*CA SiteMinder Policy Server installation Guide*
	Install the JBoss Application Server	JBoss Enterprise Application Platform documentation
	Configure the Policy Server	Preconfigure Policy Objects for the SiteMinder Agent (see page 21)
	Patch JVM for unlimited cryptography with the Java Cryptography Extension (JCE) package	Apply the Java Cryptography Patch to the JVM
	Install the SiteMinder Agent on the JBoss Enterprise Application Platform	Install the SiteMinder Agent (see page 24)
	Register system as a Trusted Host	How to Register Your System as a Trusted Host (see page 43)
	For SiteMinder Agent Security Interceptor perimeter authentication environments, install and configure additional requisite software	Additional Steps for Perimeter Authentication Installations (see page 20)

Additional Steps for SiteMinder Agent Security Interceptor Installations

Complete the steps outlined in the following table to use the SiteMinder Agent Security Interceptor to validate SiteMinder session cookies obtained by perimeter authentication.

Complete?	Steps	For information, see...
	Install a supported web server on the proxy server system.	The installation documentation provided with the web server.
	Install and configure a supported proxy module on the proxy web server.	For detailed proxy module installation and configuration directions, see the JBoss Enterprise Application Platform documentation.

Complete?	Steps	For information, see...
	Install and configure a Web Agent on the proxy server.	*CA SiteMinder Web Agent Installation Guide* *CA SiteMinder Web Agent Configuration Guide*
	Restart the web server on the proxy server.	The documentation for the web server.

Installation Location References

The following references to the installed location of SiteMinder Agent and JBoss software are used throughout this guide:

SMAGENT_HOME

Refers to the installed location of the SiteMinder Agent for JBoss.

The default location is:

- C:\Program Files\CA\JBossAgent (Windows)
- /CA/JBossAgent (UNIX)

JBOSS_HOME

Refers to the installed location of the JBoss Application Server.

For example, the default location for JBoss Enterprise Application Platform 4.3 is:

- C:\jboss-eap-4.3\jboss-as on Windows
- /jboss-eap-4.3/jboss-as on UNIX

Preconfigure Policy Objects for the SiteMinder Agent

This section describes how to preconfigure policy objects for the SiteMinder Agent for JBoss on the Policy Server.

Policy Object Preconfiguration Overview

Before you install the SiteMinder Agent for JBoss, the Policy Server must be installed and be able to communicate with the system where you plan to install the SiteMinder Agent. Additionally, configure the Policy Server with the following:

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

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

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

- **Agent object/Agent identity**

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

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

- **Host Configuration Object**

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

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

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

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

- **Agent Configuration Object**

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

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

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

Preconfigure the Policy Objects

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

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

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

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

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

   Note: If you are using SiteMinder SSO-based perimeter authentication to validate identities obtained from SiteMinder session cookies, configure separate Agents identities for the SiteMinder Agent for JBoss and the Web Agent on the proxy server.

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

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

Apply the Unlimited Cryptography Patch to the JRE

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

The files that need to be patched are:

- local_policy.jar
- US_export_policy.jar

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

- Windows

  jre_home\lib\security

- UNIX

  jre_home/lib/security

jre_home

Defines the location of your Java Runtime Environment installation.

Install the SiteMinder Agent

The following sections contain information about installing the SiteMinder Agent.

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.

Information Required During SiteMinder Agent Installation

The SiteMinder Agent for JBoss installation program prompts you to supply the following information:

- Location of the JVM to use.
- Location of the JBoss Application Server installation. For example, the default for JBoss Enterprise Application Platform 4.3 is C:\jboss-eap-4.3\jboss-as on Windows and /jboss-eap-4.3/jboss-as on UNIX.
- If you proceed to configure the Agent, the configuration wizard prompts you for the following additional information:
  - 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.
  - Agent Configuration Object name for the Agent you created when you configured the SiteMinder Policy Server for the SiteMinder agent providers

Install a SiteMinder Agent on a Windows System

The following sections describe how to install the SiteMinder Agent on a Windows system.

Set the JRE in the Path Variable

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

Follow these steps:

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

Run the Installation on Windows

Install the SiteMinder Agent for JBoss using the using the installation media on the Technical Support site.

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

Follow these steps:

1. Exit all applications that are running.
2. Navigate to where the installation executable is located.
3. Double-click ca-sm-jboss-12.52-*cr*-win32.exe.

   cr

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

   The SiteMinder Agent for JBoss installation wizard starts.
4. Use gathered system and component information to install the SiteMinder Agent. Consider the following when running the installer:
   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
5. Review the information on the Pre-Installation Summary page, then click Install.

   Note: The installation program may detect that newer versions of certain system dlls are installed on your system. It asks if you want to overwrite these newer files with older files. Select No To All if you see this message.

   The SiteMinder Agent files are copied to the specified location. Afterward, the CA SiteMinder Agent for JBoss Configuration screen is displayed.
6. Choose one of the following options:
   - Yes. I would like to configure the CA SiteMinder Agent for JBoss now.
   - No. I will configure the CA SiteMinder Agent for JBoss later.

7. Click Done.

   If you selected the option to configure the Agent now, the installation program prepares the CA SiteMinder Agent for JBoss Configuration Wizard and begins the trusted host registration and configuration processes.

   Do the following:

   - Register the trusted host. You can do this before or after configuring an Agent, but the Agent will *not* be able to communicate properly with the Policy Server unless the trusted host is registered.
   - Configure the SiteMinder Agent.

   If you did not select the option to configure the Agent now, the installation program prompts you to restart your system.Select whether to restart the system automatically or later on your own.

Installation Notes:

- After installation, you can review the installation log file in *SMAGENT_HOME*\install_config_info. The file name is: CA_SiteMinder®_Agent_for_JBoss_InstallLog.log

  SMAGENT_HOME

  Specifies the path to where the SiteMinder Agent is installed.

  Default: C:\Program Files\CA\JBossAgent

- You may choose not to start the CA SiteMinder Agent for JBoss Configuration Wizard immediately after installation or you may have to reboot your machine after installation. If so, you can start the Wizard manually when you are ready to configure an Agent.

Install the SiteMinder Agent Using the Unattended Installer on Windows

Once the SiteMinder Agent is installed on one system, you can reinstall it on the same system or install it with the same options on another system using an unattended installation mode. An unattended installation lets you install or uninstall the agent without any user interaction

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

The ca-jboss-agent-installer.properties is located in *SMAGENT_HOME*\install_config_info.

Follow these steps:

1. From a system where the agent is already installed, copy the ca-jboss-agent-installer.properties file to a local directory on your system.
2. Download the agent installation media from the Technical Support site.

   Note: For a list of installation media names for each operating system, see the installation and upgrade considerations in the *Release Notes*.
3. Copy the installation media into the same local directory as the ca-jboss-agent-installer.properties file.
4. Open a console window and navigate to the location where you copied the files.
5. Run the following command:

   ```
   ca-sm-jboss-12.52-cr-win32.exe  -f ca-jboss-agent-installer.properties -i
   silent
   ```

   cr

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

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

   When running this command, if the ca-jboss-agent-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   For example:

   ```
   ca-sm-jboss-12.52-cr-win32.exe -f "C:\Program
   Files\CA\JBossAgent\install_config_info\ca-jboss-agent-installer.properties "
   -i silent
   ```

   An InstallAnywhere status bar appears, which shows that the unattended SiteMinder Agent installer has begun. The installer uses the parameters specified in the ca-jboss-agent-installer.properties file.

Note: To stop the installation manually, open the Windows Task Manager and stop the *installation_media* process.

To verify that the unattended installation completed successfully, see the CA_SiteMinder®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*\install_config_info directory. This log file contains the results of the installation.

Install a SiteMinder Agent on a UNIX System

The following sections describe how to install the SiteMinder Agent on a UNIX system.

Set the JRE in the PATH Variable

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

To set the JRE in the PATH variable

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

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

   JRE

   Defines the location of your Java Runtime Environment bin directory.

Run the Installer in GUI Mode on UNIX

Install the SiteMinder Agent for JBoss using the installation media on the Technical Support site.

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

Follow these steps:

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located
3. If necessary, add executable permissions to the install file by running the following command:

   ```
   chmod +x installation_media
   ```

   installation_media

   Specifies the SiteMinder Agent installer executable
4. Enter the following command:

   ```
   sh ./ca-sm-jboss-12.52-cr-unix_version.bin
   ```

   cr

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

   unix_version

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

   The SiteMinder Agent for JBoss installation wizard starts.

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

   - If you enter path information in the wizard by cutting and pasting, enter (and delete, if necessary) at least one character to enable the Next button.
   - When prompted to select the Java version, the installer lists all Java executables present on the system. Select a supported 32-bit Java Runtime Environment (refer to the Platform Support Matrix on the Technical Support site).
   - Do not use space characters in the SiteMinder WSS Agent install path. For example, "/CA Technologies/agent" will result in install failure.

6. Review the information displayed on the Pre-Installation Summary page, then click Install.

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

   The SiteMinder Agent files are copied to the specified location. Afterward, the CA SiteMinder Agent for JBoss Configuration screen is displayed.

7. Choose one of the following options:

   - Yes. I would like to configure the CA SiteMinder Agent for JBoss now.
   - No. I will configure the CA SiteMinder Agent for JBoss later.

8. Click Done.

   If you selected the option to configure the Agent now, the installer prepares the CA SiteMinder Agent for JBoss Configuration Wizard and begins the host registration and configuration processes.

   Do the following:

   - Register the trusted host. You can perform this process before or after configuring an Agent. However the Agent *cannot* communicate properly with the Policy Server unless the trusted host is registered.
   - Configure the SiteMinder Agent.

   If you did not select the option to configure the Agent now, the installation program prompts you to restart your system. Select whether to restart the system automatically or later on your own.

Installation Notes:

- After installation, you can review the installation log file in *SMAGENT_HOME/*install_config_info. The file name is: CA_SiteMinder®_Agent_for_JBoss_InstallLog.log

  SMAGENT_HOME

  Specifies the path to where the SiteMinder Agent is installed.

- If you do not start the configuration wizard immediately after installation, you can start the Wizard manually when you are ready to configure an Agent.
- If you must reboot the server after installation, you can start the Wizard manually when you are ready to configure an Agent.

Run the Installer in Console Mode on UNIX

Install the SiteMinder Agent for JBoss using the installation media on the Technical Support site.

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

Follow these steps:

1. Exit all applications that are running.
2. Open a shell and navigate to where the install program is located
3. If necessary, add executable permissions to the install file by running the following command:

   ```
   chmod +x installation_media
   ```

   installation_media

   Specifies the SiteMinder Agent installer executable

4. Enter the following command:

   ```
   sh ./ca-sm-jboss-12.52-cr-unix_version.bin -i console
   ```

   cr

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

   unix_version

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

   The SiteMinder Agent for JBoss installation wizard starts.

   The SiteMinder Agent for JBoss installation wizard starts.

5. Use gathered system and component information to install the SiteMinder Agent. Consider the following as you make your selections:

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

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

6. Review the information displayed on the Pre-Installation Summary page, then proceed.

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

   The SiteMinder Agent files are copied to the specified location. Afterward, the CA SiteMinder Agent for JBoss Configuration page is displayed.

7. Select whether to restart the system now or later on your own.

8. Hit Enter.

Note: After installation, you can review the installation log file in *SMAGENT_HOME/*install_config_info. The file name is: CA_SiteMinder®_Agent_for_JBoss_InstallLog.log.

Install the SiteMinder Agent Using the Unattended Installer on UNIX

Once the SiteMinder Agent is installed on one system, you can reinstall it on the same system or install it with the same options on another system using an unattended installation mode. An unattended installation lets you install or uninstall the agent without any user interaction

The unattended installation uses the ca-jboss-agent-installer.properties file generated during the initial install from the information you specified to define the necessary installation parameters, passwords, and so on.The ca-jboss-agent-installer.properties is located in *SMAGENT_HOME/*install_config_info.

Follow these steps:

1. From a system where the SiteMinder Agent is already installed, copy the ca-jboss-agent-installer.properties file to a local directory on your system.

2. Download the agent installation media from the Technical Support site.

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

3. Copy the installation media into the same local directory as the ca-jboss-agent-installer.properties file.

4. Open a console window and navigate to the location where you copied the files.

5. Run the following command:

   ```
   ca-sm-jboss-12.52-cr-unix_version.bin -f ca-jboss-agent-installer.properties
   -i silent
   ```

   cr

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

   unix_version

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

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

   When running this command, if the ca-jboss-agent-installer.properties file is not in the same directory as the installation program, use double quotes if the argument contains spaces.

   For example:

   ca-sm-jboss-12.52-*cr-unix_version*.bin -f "/CA/JBossAgent/install_config_info/ca-jboss-agent-installer.properties " -i silent

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

   An InstallAnywhere status bar appears, which shows that the unattended SiteMinder Agent installer has begun. The installer uses the parameters specified in the ca-jboss-agent-installer.properties file.

Note: To stop the installation manually, type Ctrl+C.

To verify that the unattended installation completed successfully, see the CA_SiteMinder®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME/*install_config_info directory. This log file contains the results of the installation.

Configure the JVM to Use the JSafeJCE Security Provider

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

Follow these steps:

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

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

   JVM_HOME

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

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

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

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

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

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

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

How to Configure the Agent and Register A System as a Trusted Host on Windows

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

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

Gather Information Required for SiteMinder WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

SM Admin User Name

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

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

SM Admin Password

The Policy Server administrator account password.

Trusted Host Name

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

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

Host Configuration Object

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

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

Policy Server IP Address

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

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

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

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

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

FIPS Encryption Mode

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

FIPS Compatibility Mode (Default)

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

FIPS Only Mode

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

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

Configure Agents and Register Your System as a Trusted Host

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

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

To configure Agents and register a trusted host

1. If necessary, start the SiteMinder Configuration Wizard. The default method is to select Start, Programs, CA, SiteMinder, SiteMinder Configuration Wizard. If you have placed the Wizard shortcut in a non-default location, the procedure will be different.

   (Alternatively, navigate to *SMAGENT_HOME*\install_config_info and run ca-jbossagent-config.exe.)

   Note: If you chose to configure the SiteMinder Agent immediately after the installation, the installer automatically starts the Configuration Wizard.

   The SiteMinder Configuration Wizard starts.

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

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

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

Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of the SiteMinder Agent for JBoss, check the CA_SiteMinder®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*/install_config_info directory.

Modify the SmHost.conf File

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

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

To modify the SmHost.conf file

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

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

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

   hostconfigobject

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

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

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

   policyserver

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

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

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

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

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

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

   Default: *IP_address*, 44441,44442,44443

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

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

requesttimeout

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

Default: 60

Example: requesttimeout="60"

4. Save and close the SmHost.Conf file.

   The changes to the SmHost.conf file are applied.

Re-register a Trusted Host Using the Registration Tool

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

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

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

To re-register a trusted host using the registration tool

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

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

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

See the following example:

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

The following example contains the -o argument:

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

The following arguments are used with the smreghost command:

-i *policy_server_IP_ address:port*

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

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

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

Default: (ports) 44441,44442,44443

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

Example: (IPv4 default ports) -i 127.0.0.1

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

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

-u *administrator_username*

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

-p *Administrator_password*

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

-hn *hostname_for_registration*

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

-hc *host_config_object*

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

-sh *shared_secret*

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

-rs

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

-f *path_to_host_config_file*

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

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

-cf *FIPS mode*

Specifies one of the following FIPS modes:

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

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

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

Default: COMPAT

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

-o

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

The trusted host is re-registered.

Register Multiple Trusted Hosts on One System

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

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

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

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

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

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

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

How to Configure the Agent and Register a System as a Trusted Host on UNIX

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

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

Gather Information Required for SiteMinder WSS Agent Configuration

The following information must be supplied during Trusted Host registration:

SM Admin User Name

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

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

SM Admin Password

The Policy Server administrator account password.

Trusted Host Name

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

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

Host Configuration Object

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

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

Policy Server IP Address

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

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

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

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

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

FIPS Encryption Mode

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

FIPS Compatibility Mode (Default)

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

FIPS Only Mode

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

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

Configure Agents and Register a Trusted Host in GUI or Console Mode

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

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

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

- You may be instructed to select an option by entering a corresponding number for that option.
- You press Enter after each step to proceed through the process. The prompts should guide you through the process.
- All passwords that you enter are displayed in clear text. To workaround this issue, run the installation in GUI or unattended mode.

To configure Agents and register a trusted host

1. If necessary, start the Configuration Wizard as follows:
   1. Open a console window.
   2. Navigate to *SMAGENT_HOME*/install_config_info, where *agent_home* is the installed location of the SiteMinder Agent.
   3. Enter one of the following commands:

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

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

   The Configuration Wizard starts.
2. Use gathered system and component information to configure the SiteMinder Agent and register the host.

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

Installation and Configuration Log Files

To check the results of the installation or review any specific problems during the installation or configuration of the SiteMinder Agent for JBoss, check the CA_SiteMinder®_Agent_for_JBoss_InstallLog.log file in the *SMAGENT_HOME*/install_config_info directory.

Modify the SmHost.conf File

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

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

To modify the SmHost.conf file

1. Navigate to the *SMAGENT_HOME/*config directory.
2. Open the SmHost.conf file in a text editor.
3. Enter new values for the any of the following settings that you want to change:

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

   hostconfigobject

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

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

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

   policyserver

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

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

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

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

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

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

Default: *IP_address*, 44441,44442,44443

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

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

requesttimeout

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

Default: 60

Example: requesttimeout="60"

4. Save and close the SmHost.Conf file.

   The changes to the SmHost.conf file are applied.

Re-register a Trusted Host Using the Registration Tool

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

- To rename the trusted host if there has been a change to your CA SiteMinder® environment.
- To register a trusted host if the trusted host has been deleted in the Administrative UI.
- To register a trusted host if the trusted host policy objects have been deleted from the policy store or the policy store has been lost.

- To change the shared secret that secures the connection between the trusted host and the Policy Server.
- To recreate the SmHost.conf configuration file if it is lost.
- To overwrite an existing trusted host without deleting it first.

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

To re-register a trusted host using the registration tool

1. Open a command prompt window.
2. Ensure that the library path environment variable contains the path to the SiteMinder Agent's bin directory by entering the following two commands:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH:agent_home/bin
   ```

   ```
   export LD_LIBRARY_PATH
   ```

   For example:

   ```
   LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/CA/JBossAgent/bin
   ```

   ```
   export LD_LIBRARY_PATH
   ```

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

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

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

   See the following example:

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

   The following example contains the -o argument:

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

The following arguments are used with the smreghost command:

-i *policy_server_IP_ address:port*

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

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

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

Default: (ports) 44441,44442,44443

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

Example: (IPv4 default ports) -i 127.0.0.1

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

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

-u *administrator_username*

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

-p *Administrator_password*

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

-hn *hostname_for_registration*

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

-hc *host_config_object*

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

-sh *shared_secret*

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

-rs

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

-f *path_to_host_config_file*

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

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

-cf *FIPS mode*

Specifies one of the following FIPS modes:

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

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

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

Default: COMPAT

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

-o

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

The trusted host is re-registered.

Register Multiple Trusted Hosts on One System

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

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

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

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

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

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

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

Uninstall a SiteMinder Agent for JBoss

To uninstall a SiteMinder Agent, run the SiteMinder uninstall wizard.

To uninstall the SiteMinder Agent on Windows or UNIX systems

1. Navigate to the *SMAGENT_HOME*\install_config_info (Windows) or *SMAGENT_HOME*/install_config_info (UNIX) directory and run the SiteMinder uninstall wizard:

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

   The uninstall wizard starts.

2. Confirm that you want to remove the SiteMinder Agent.

The uninstall wizard removes the SiteMinder Agent.

Note: You may also want to revert any JBoss configuration files that you modified for the SiteMinder Agent to their previous state.

Chapter 3: Configure the SiteMinder Agent

This section contains the following topics:

SiteMinder Agent for JBoss Configuration File

By default, the SiteMinder Agent for JBoss installation creates a single agent configuration file, JavaAgent.conf in the *SMAGENT_HOME/*config directory.

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

Parameter	Description
DefaultAgentName	The agent identity the Policy Server uses to associate policies with the SiteMinder Agent.
EnableAgent	Specifies whether the SiteMinder Agent is enabled. Possible values are Yes and No. Default value is Yes.
AgentConfigObject	The Agent Configuration Object specified during installation.
SmHostFile	Path to the local Host Configuration File. Path can be specified in absolute terms or relative to *SMAGENT_HOME.* **Note:** On Windows, you must specify paths using double backslashes ("\\") rather than single backslash ("\") to separate directories. On UNIX, use standard single slash ("/") separators. Example values: ■ (Windows) C:\\Program Files\\CA\\JBossAgent\\config\\SmHost.conf ■ (Windows) config\\SmHost.conf ■ (UNIX) export/JBossAgent/config/SmHost.conf ■ (UNIX) /config/SmHost.conf
ServerName	A string that will be used in the SiteMinder Agent log to identify the JBoss Application Server.

Parameter	Description
appserverjaasloginhandler	Specifies the SiteMinder Agent for JBoss login handler class. Default value is "com.ca.soa.agent.appserver.jaas.jboss.JBossLoginHandler". Do not change this value.
appserverjmshandler	Specifies the SiteMinder Agent for JBoss JMS handler class. Default value is "com.ca.soa.agent.appserver.jaxrpc.jms.jboss.JBossJMSMessageHandler". Do not change this value.

You should not need to edit the preconfigured values unless the location of the Host Configuration File changes or you want to refer to a different Agent Configuration Object. If you choose to use local configuration, you can add other Agent configuration parameters to these preconfigured values.

Note: Parameters held in the Agent configuration file are static; if you change these settings while the JBoss server is running, the SiteMinder Agent will not pick up the change until JBoss is restarted.

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

Generally, you only need to edit the JavaAgent.conf.file if you change the name of your Agent Configuration Object.

Sample JavaAgent.conf (Windows)

```
# Java Agent Configuration File
#
# This file contains bootstrap information required by
# the SiteMinder Java Agent
#

#
# Configuration for agent testagent
#
defaultagentname=agentjboss
enablewebagent=yes
agentconfigobject=soaagentconfig
servername=jboss.example.com
smhostfile=C:\\Program Files\\CA\\JBossAgent\\config\\SmHost.conf

appserverjaasloginhandler=com.ca.soa.agent.appserver.jaas.jboss.JBossLoginHandler
appserverjmshandler=com.ca.soa.agent.appserver.jaxrpc.jms.jboss.JBossJMSMessageHa
ndler

# Configure plugins for the agent testagent
transport_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig,
com.ca.soa.agent.jaxrpcplugin.pluginconfig.JaxRpcPluginConfig,
com.ca.soa.agent.jmsplugin.pluginconfig.JMSPluginConfig,
com.ca.soa.agent.jaxwsplugin.pluginconfig.JaxWsPluginConfig
msg_body_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginConfig,
com.ca.soa.agent.jaxwsplugin.pluginconfig.JaxWsPluginConfig
credential_plugin_list=com.ca.soa.agent.httpplugin.pluginconfig.HttpPluginConfig,
com.ca.soa.agent.jaxwsplugin.pluginconfig.JaxWsPluginConfig
variable_resolver_plugin_list=com.ca.soa.agent.txmplugin.pluginconfig.TxmPluginCo
nfig

# <EOF>
```

Agent Configuration Object

An Agent Configuration Object is a Policy Server object that holds Agent parameters for an Agent when using central agent configuration.

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

SiteMinder Agent Configuration Parameters

The following table contains a complete list of all Agent configuration parameters supported by the SiteMinder Agent for JBoss.

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

Parameter Name	Value	Description
AcceptTPCookie	YES or NO	(Optional) If set to yes, configures the SiteMinder Agent to assert identities from third-party SiteMinder session cookies (that is, session cookies generated by custom Agents created using the SiteMinder and CA SiteMinder® Web Services Security SDKs. **Note:** AcceptTPCookie must be set to Yes to assert identities from session cookies generated by CA SOA Security Gateway. Default is Yes.
AgentName	String	Defines the identity of the SiteMinder Agent. It establishes a mapping between the name and the IP address of each web server instance hosting an Agent. If a value is not set for this parameter, or if the SiteMinder Agent does not find a match among the values listed, the SiteMinder Agent uses the value set in the DefaultAgentName parameter instead. **Note:** This parameter can have more than one value. Use the multi-value option when setting this parameter in an Agent Configuration Object. For local configuration files, add the parameter name followed by each value to separate lines in the file. No default value.
AllowLocalConfig (Applies only in the Agent Configuration Object)	YES or NO	If set to yes, parameters set locally in the Agent configuration file take precedence over parameters in the Agent Configuration Object. Default is NO.

Parameter Name	Value	Description
AuthCacheSize	Number	(Optional) Size of the authentication cache for the SiteMinder Agent (in number of entries). For example: `authcachesize="1000"` Default is 0. To flush this cache, use the Policy Server User Interface.
AzCacheSize		(Optional) Size of the authorization cache (in number of entries) for the SiteMinder Agent. For example: `authcachesize="1000"` Default is 0. To flush this cache, use the Policy Server User Interface.
CacheTimeout	Number	(Optional) Number of seconds before cache times out. For example: `cachetimeout="1000"` Default is 600 (10 minutes).
ConfigObject (Applies only in Agent configuration file)	String	The name of the Agent Configuration Object associated with the SiteMinder Agent. No default value.
CookieDomain	String	(Optional) Name of the cookie domain. For example: `cookiedomain="ca.com"` No default value. For more information, see the cookiedomainscope parameter.

Parameter Name	Value	Description
CookieDomainScope	Number	(Optional) Further defines the cookie domain for assertion of SiteMinder session cookies by the SiteMinder Agent. The scope determines the number of sections, separated by periods, that make up the domain name. A domain always begins with a period (.) character. For example: `cookiedomainscope="2"` Default is 0, which takes the domain name specified in the cookiedomain parameter.
DefaultAgentName (Applies only in the Agent Configuration Object)	String	The agent identity the Policy Server will use to associate policies with the SiteMinder Agent if there is no agent name specified in the AgentName parameter. No default value.
EnableWebAgent (Applies only in Agent configuration file)	YES or NO	Enables or disables the SiteMinder Agent. When set to 'yes', the SiteMinder Agent will protect resources using the Policies configured in the Policy Server for the configured agent identity. Default is Yes.
LogOffUri	String	(Optional) The URI of a custom HTTP file that will perform a full log off (removing the session cookie from a user's browser). A fully qualified URI is not required. For example, LogOffUri could be set to: /Web pages/logoff.html No default value.
PsPollInterval	Number	(Optional) The frequency with which the SiteMinder Agent polls the Policy Server to retrieve information about policy changes. Default is 30 seconds.
ResourceCacheSize	Number	(Optional) Size (in number of entries) of the cache for resource protection decisions. For example: `resourcecachesize="1000"` Default is 2000. To flush this cache, use the Administrative UI.

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

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

Chapter 4: Configure JBoss to Work with the SiteMinder Agent

This section contains the following topics:

Configure Agent-related Environment Settings on JBoss 5.x

To configure the agent to operate with the JBoss 5.x Application Server, complete one of the following procedures:

- Set the JBoss 5.x Environment on Windows (see page 61).
- Set the JBoss 5.x Environment on UNIX (see page 62).

Set the JBoss 5.x Environment on Windows

Before the SiteMinder Agent can operate with the JBoss Application Server, you must configure SiteMinder Agent-related environment settings on Windows by editing the JBoss run.bat script.

To configure SiteMinder Agent-related environment settings

1. Navigate to the *JBOSS_HOME*\bin directory
2. Open the run.bat file in a text editor.
3. Add the following entry to specify the installed location of the SiteMinder Agent for JBoss

   ```
   set SOA_HOME=SMAGENT_HOME
   ```

4. Add the following entry to define required JVM system properties for the agent:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -DJAVA_AGENT_ROOT=%SOA_HOME%
   -Dlog.log-config-properties=%SOA_HOME%\config\log-config.properties
   -Dfile.encoding=UTF8
   ```

5. Add the following entry to include directories required for SiteMinder Agent operation in the JBOSS_CLASSPATH:

   ```
   set
   JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%SOA_HOME%\config;%JBOSS_HOME%\server\defau
   lt\lib\cryptojFIPS.jar
   ```

6. By default, JBoss only listens for requests on the localhost IP address. To configure JBoss to listen on all IP addresses, locate the entry following the remark line "Execute the JVM in the background" and change "org.jboss.Main" to "org.jboss.Main -b 0.0.0.0". For example:

   ```
   "%JAVA%" %JAVA_OPTS% -Djava.endorsed.dirs="%JBOSS_ENDORSED_DIRS%"
   -classpath "%JBOSS_CLASSPATH%" org.jboss.Main -b 0.0.0.0 %*
   ```

7. Save your changes.

8. Restart the JBoss Application Server to apply the changes.

Set the JBoss 5.x Environment on UNIX

Before the SiteMinder Agent can operate with the JBoss Application Server, you must configure SiteMinder Agent-related environment settings on UNIX by editing the JBoss run.sh script.

To configure SiteMinder Agent-related environment settings

1. Navigate to the *JBOSS_HOME*/bin directory

2. Open the run.sh file in a text editor.

3. Add the following lines to specify the installed location of the SiteMinder Agent for JBoss:

   ```
   SOA_HOME=SMAGENT_HOME
   export SOA_HOME
   ```

4. Add the following entry to define required JVM system properties for the agent:

   ```
   JAVA_OPTS=$JAVA_OPTS -DJAVA_AGENT_ROOT=$SOA_HOME
   -Dlog.log-config-properties=$SOA_HOME/config/log-config.properties
   -Dfile.encoding=UTF8
   export JAVA_OPTS
   ```

5. Add the following entry to include directories required for SiteMinder Agent operation in the JBOSS_CLASSPATH:

   ```
   JBOSS_CLASSPATH=$JBOSS_CLASSPATH:$SOA_HOME/config:$JBOSS_HOME/server/default/
   lib/cryptojFIPS.jar
   export JBOSS_CLASSPATH
   ```

6. By default, JBoss only listens for requests on the localhost IP address. To configure JBoss to listen on all IP addresses, locate the entry following the remark line "Execute the JVM in the background" and change "org.jboss.Main" to "org.jboss.Main -b 0.0.0.0". For example:

   ```
   "$JAVA" $JAVA_OPTS -Djava.endorsed.dirs="$JBOSS_ENDORSED_DIRS"
   -classpath "$JBOSS_CLASSPATH" org.jboss.Main -b 0.0.0.0 *
   ```

7. Save your changes.

8. Restart the JBoss Application Server to apply the changes.

Configure Agent-related Environment Settings on JBoss 6.x

To configure the agent to operate with a JBoss 6.x Application Server, complete one of the following procedures:

- Set the JBoss 6.x Environment on Windows (see page 63).
- Set the JBoss 6.x Environment on UNIX (see page 63).

Set the JBoss 6.x Environment on Windows

Configure agent-related environment settings on Windows by editing the standalone.conf.bat script.

Follow these steps:

1. Navigate to the *JBOSS_HOME*\bin directory
2. Open the standalone.conf.bat file in a text editor.
3. Add the following entry to specify the installed location of the SiteMinder Agent for JBoss

   ```
   set SOA_HOME=SMAGENT_HOME
   ```

4. Add the following entry to define required JVM system properties for the agent:

   ```
   set "JAVA_OPTS=%JAVA_OPTS% -DJAVA_AGENT_ROOT=%SOA_HOME%
   -DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true
   -DTXM_DOCUMENT_BUILDER=org.apache.xerces.jaxp.DocumentBuilderFactoryImpl"
   ```

5. Save your changes.
6. Restart the JBoss Application Server to apply the changes.

Set the JBoss 6.x Environment on UNIX

Configure agent-related environment settings on UNIX by editing the JBoss standalone.conf script.

Follow these steps:

1. Navigate to the *JBOSS_HOME*/bin directory
2. Open the standalone.conf file in a text editor.
3. Add the following lines to specify the installed location of the SiteMinder Agent for JBoss:

   ```
   SOA_HOME=SMAGENT_HOME
   ```

4. Add the following entry to define required JVM system properties for the agent:

   ```
   JAVA_OPTS="$JAVA_OPTS -DJAVA_AGENT_ROOT=$SOA_HOME
   -DSM_AGENT_LOGGING_EXTERNAL_CONFIG=true
   -DTXM_DOCUMENT_BUILDER=org.apache.xerces.jaxp.DocumentBuilderFactoryImpl"
   ```

5. Save your changes.

6. Restart the JBoss Application Server to apply the changes.

Chapter 5: Configure SiteMinder Agent Logging

This section contains the following topics:

Logging Overview

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

Two log files provide important information about the SiteMinder Agent:

SiteMinder Agent logging

The agent writes information about its standard operations and performance such as error and processing messages to the SiteMinder Agent log.

SiteMinder Agent XML message processing logging file

In addition to its standard logging functionality, the agent also logs information relating specifically to WSS Agent Security Interceptor XML message processing.

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

Configure SiteMinder Agent Logging on JBoss 5.x

By default, SiteMinder Agent logging is enabled and written to the XmlAgent.log file in one of the following locations:

- Windows—*JBOSS_HOME*\bin\soa-log\XmlAgent.log
- UNIX—*JBOSS_HOME*/bin/soa-log/XmlAgent.log

Change SiteMinder Agent logging parameters by editing the log-config.properties file located in one of the following locations:

- Windows—*SMAGENT_HOME*\config\
- UNIX— *SMAGENT_HOME*/config/

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

Available logging parameters are as follows:

Name	Description
log.logfile-append-on-reset	Add logging information to an existing log file instead of creating a new file each time logging is invoked. Default value: no
log.logfile-pattern	Specifies the pathname (relative to *JBOSS_HOME*/bin) of the SiteMinder Agent log file. Default value: /soa-log/XmlAgent.log
log.logging-level	Defines the logging level. The levels are: ■ DEBUG - all logging, most verbose ■ CONFIG - configuration information ■ INFO - information ■ WARN -warnings ■ SEVERE - errors Default value: SEVERE
log.logfile-limit	Specifies the size limit, in kilobytes Rollover a log file after it reaches the specified size. Default value: 1000

Configure SiteMinder Agent XML Message Processing Logging on JBoss 5.x

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

- Windows—*SMAGENT_HOME*\bin\
- UNIX—*SMAGENT_HOME*/bin/

Change SiteMinder Agent XML message processing logging parameters by editing the log.config file, located in one of the following directions:

- Windows—*SMAGENT_HOME*\config\
- UNIX— *SMAGENT_HOME*/config/

Configure Logging on JBoss 6.x

To configure logging on JBoss 6.x, edit the standalone.xml file that is located in one of the following locations:

- **Windows**: *JBOSS_HOME*\standalone\configuration
- **UNIX**: *JBOSS_HOME*/standalone/configuration

In a text editor, add the following text to the logging subsystem section to configure logging with recommended default values:

```
<size-rotating-file-handler name="AgentFile" autoflush="true">
    <formatter>
      <pattern-formatter pattern="%d %-5p [%c] (%t) %m%n"/>
    </formatter>
    <file path="AGENT_HOME/log/XmlAgent.log"/>
    <rotate-size value="1000k"/>
    <max-backup-index value="5"/>
    <append value="true"/>
</size-rotating-file-handler>
<periodic-rotating-file-handler name="SOASMAgentFile" autoflush="true">
     <formatter>
          <pattern-formatter pattern="%d [%p] %c{3} %x - %m%n"/>
     </formatter>
     <file path="AGENT_HOME/log/soasm_agent.log"/>
           <suffix value=".yyyy-MM-dd"/>
          <append value="true"/>
  </periodic-rotating-file-handler>
```

```
<logger category="com.ca.soa" use-parent-handlers="false">
  <level name="INFO"/>
  <handlers>
    <handler name="AgentFile"/>
  </handlers>
</logger>
<logger category="com.netegrity.tm" use-parent-handlers="false">
  <level name="INFO"/>
  <handlers>
    <handler name="SOASMAgentFile"/>
  </handlers>
</logger>
```

You can change the values of the following configurable logging parameters:

<file path> (first instance)

Specifies the pathname of the SiteMinder Agent log file.

<rotate-size value>

Specifies the size limit, in kilobytes before the SiteMinder Agent log file rolls over.

<append value>

Specifies whether logging information is added to an existing log file instead of creating a file each time that logging is invoked. Specify one of the following values:

- true
- false

<file path> (second instance)

Specifies the pathname of the XML message processing log file.

<level name> (first instance)

Defines the SiteMinder Agent logging level. Specify one of the following values:

- ALL
- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF

<level name> (second instance)

Defines the XML message processing logging level. Specify one of the following values:

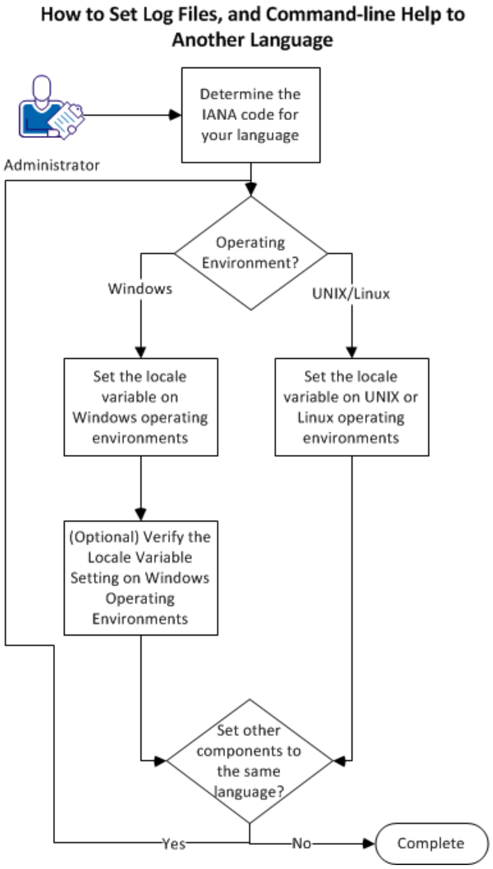
- ALL
- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF

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

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

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

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


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

Follow these steps:

1. Determine the IANA code for your language (see page 71).
2. Create the environment variable for your operating environment using one of the following procedures:
   - Set the locale variable on Windows operating environments (see page 72).
   - Set the locale variable on UNIX or Linux operating environments (see page 74).
3. (Optional) Verify the locale variable setting on windows operating environments (see page 73).
4. (Optional) Repeat Steps 1 through 3 to set any other components in your environment to the same language.

Determine the IANA Code for Your Language

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

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

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

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

Environment Variables

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

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

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

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

- Set the locale variable on Windows operating environments (see page 72).
- Set the locale variable on UNIX or Linux operating environments (see page 74).

Set the Locale Variable on Windows Operating Environments

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

```
SM_ADMIN_LOCALE
```

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

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

Follow these steps:

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

   The system properties dialog appears.

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

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

5. Type the following text:

   ```
   SM_ADMIN_LOCALE
   ```

6. Click the Variable name: field, and then type the IANA language code (see page 71) that you want.
7. Click OK.

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

8. Click OK *twice*.

   The locale variable is set.

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

Verify the Locale Variable Value on Windows Operating Environments

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

Note: Instructions for verifying the variable value on UNIX and Linux are in the setting procedure (see page 74).

Follow these steps:

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

   a. Click Start, Run.

   b. Type the following command:

   ```
   cmd
   ```

   c. Click OK.

   A command-line window opens.

2. Enter the following command:

   ```
   echo %SM_ADMIN_LOCALE%
   ```

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

   ```
   de
   ```

   The value of the locale variable is verified.

Set the Locale Variable on UNIX or Linux Operating Environments

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

```
SM_ADMIN_LOCALE
```

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

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

Follow these steps:

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

   ```
   export SM_ADMIN_LOCALE=IANA_language_code
   ```

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

   ```
   export SM_ADMIN_LOCALE=fr
   ```

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

   ```
   echo $SM_ADMIN_LOCALE
   ```

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

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

Chapter 6: Configure the SiteMinder Agent Security Interceptor to Protect Web Applications on JBoss 5.x

This section contains the following topics:

Configure SiteMinder Agent Authenticators

SiteMinder Agent Authenticators extend the functionality of the JBossSX default authenticators with the ability to authenticate a user request based on an associated SiteMinder session cookie.

You can configure the SiteMinder Agent Authenticators into the JBoss security infrastructure for all web applications or for individual web applications as required.

Configure SiteMinder Agent Authenticators For All Web Applications on JBoss 5.x

To configure the SiteMinder Agent Authenticators to handle all JBoss web application requests, replace the default JBossSX authenticator methods with the SiteMinder Agent Authenticator methods in the JBoss core authentication services definition.

The JBoss core authentication services are defined in the war-deployers-jboss-beans.xml configuration file located in the following location:

```
server/server_name/deployers/jbossweb.deployer/META-INF
```

Note: The SiteMinder Agent Authenticator methods extend the default authenticator methods; the default authenticator functionality is still available for requests without valid SiteMinder session cookies.

To Configure SiteMinder Agent Authenticators at the global level

1. Navigate to server/*server_name*/deployers/jbossweb.deployer/META-INF.
2. Open the war-deployers-jboss-beans.xml file in a text editor.

3. Locate the <attribute name="Authenticators" ...> element definition section.

4. Edit the java:value element in the java:property element definitions for BASIC, FORM, CLIENT-CERT, and DIGEST authentication, replacing the default authenticator methods with the corresponding SiteMinder Agent Authenticator methods as required.

   To configure the SMJBossBasicAuthenticator, edit the java:property element for BASIC authentication as follows:

   ```
   <entry>
     <key>BASIC</key>

   <value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossBasicAuthenticat
   or</value>
   </entry>
   ```

   To configure the SMJBossFormAuthenticator, edit the java:property element for FORM authentication as follows:

   ```
   <entry>
     <key>FORM</key>

   <value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossFormAuthenticato
   r</value>
   </entry>
   ```

   To configure the SMJBossClientCertAuthenticator, edit the java:property element for CLIENT-CERT authentication as follows:

   ```
   <entry>
     <key>CLIENT-CERT</key>

   <value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossClientCertAuthen
   ticator</value>
   </entry>
   ```

   To configure the SMJBossDigestAuthenticator, edit the java:property element for DIGEST authentication as follows:

   ```
   <entry>
     <key>DIGEST</key>

   <value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossDigestAuthentica
   tor</value>
   </entry>
   ```

If you do not want the default authentication behavior to occur if SiteMinder session cookie validation fails, configure the SMJBossIdentityAsserter in place of any authenticator. For example, to configure the SMJBossIdentityAsserter so that default Digest authentication does not occur if SiteMinder identity assertion fails, edit the java:property element for DIGEST as follows:

```
<entry>
  <key>DIGEST</key>

<value>com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossIdentityAsserter
</value>
</entry>
```

5. Save the file and exit the text editor.

The SiteMinder Agent Authenticators are configured as the default authenticators for all security-enabled web applications. The authenticator configured for the authentication method defined in the web application deployment descriptor will handle request unless an authenticator is configured individually for that application.

Configure a SiteMinder Agent Authenticator for an Individual Application on JBoss 5.x

To configure a web application to use a specific SiteMinder Agent Authenticator to handle requests, define a context.xml file in the application WEB-INF directory. Configuring a context.xml file overrides the global authenticators defined in war-deployers-jboss-beans.xml.

To configure a web application to use a specific SiteMinder Agent Authenticator

1. Navigate to the application WEB-INF directory.
2. Open a text editor.
3. Define a context element containing a valve subelement that specifies the class name of the SiteMinder Agent Authenticator which you want to handle application requests.

   To configure the application to use SMJBossBasicAuthenticator, type:

```
<Context cookies="true" crossContext="true">
  <Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossBasicAuthent
icator"/>
</Context>
```

To configure the application to use the SMJBossFormAuthenticator, type:

```
<Context cookies="true" crossContext="true">
  <Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossFormAuthenti
cator"/>
</Context>
```

To configure the application to use SMJBossClientCertAuthenticator, type:

```
<Context cookies="true" crossContext="true">
  <Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossClientCertAu
thenticator"/>
</Context>
```

To configure the application to use SMJBossDigestAuthenticator, type:

```
<Context cookies="true" crossContext="true">
  <Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossDigestAuthen
ticator"/>
</Context>
```

To configure the application to use the SMJBossIdentityAsserter, type:

```
<Context cookies="true" crossContext="true">
  <Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossIdentityAsse
rter"/>
</Context>
```

4. Save the file as context.xml and exit the text editor.

Define a JBossSX Security Domain for the SiteMinder Agent Login Module

Define a JBoss security domain named SiteMinderDomain that configures the SiteMinder Agent Login Module required to authenticate credentials obtained by SiteMinder Agent authenticators. Configure the SiteMinderDomain by adding an application-policy element to the login-config.xml file located in server/*server_name*/conf/.

To configure SiteMinder Agent Authenticators at the global level

1. Navigate to server/*server_name*/conf/login-config.xml
2. Open the login-config.xml file in a text editor.

3. Add the following application-policy element defining the SiteMinderDomain:

```
<application-policy name="SiteMinderDomain">
  <authentication>
    <login-module
    code="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossLoginModule"
    flag="required">
    <module-option name="unauthenticatedIdentity">anonymous</module-option>
    </login-module>
  </authentication>
</application-policy>
```

4. Save the file and exit the text editor.

Configure Web Applications to Invoke the SiteMinder Agent Security Interceptor

To protect a web application using the SiteMinder Agent Security Interceptor, edit its deployment descriptor to enable security and map it to the SiteMinderDomain security domain.

Edit the Application Deployment Descriptor to Enable Security

Edit the web.xml deployment descriptor to enable security for each web application that you want to protect with the SiteMinder Agent Web Interceptor. The web.xml file is located in the application WEB-INF directory.

For more information about the web.xml file and constituent element syntax, see the JBoss Enterprise Application Platform documentation.

To Edit the web.xml deployment descriptor to enable security

1. Navigate to the web application WEB-INF directory

2. Open the web.xml deployment descriptor file in a text editor.

3. Add one or more security-constraint elements defining what resources in the web application are to be protected. For example:

   ```
   <security-constraint>
       <display-name>Constraint1</display-name>
       <web-resource-collection>
           <web-resource-name>admin resource</web-resource-name>
           <description/>
           <url-pattern>/admin/*</url-pattern>
           <http-method>GET</http-method>
           <http-method>POST</http-method>
       </web-resource-collection>
       <auth-constraint>
           <description/>
           <role-name>adminRole</role-name>
       </auth-constraint>
   </security-constraint>
   ```

4. Add a security-role element defining roles used by the application. For example:

   ```
   <security-role>
        <description/>
        <role-name>adminRole</role-name>
   </security-role>
   ```

5. Add a login-config element. The auth-method subelement of the login-config element defines the authentication method (BASIC, FORMS, and so on) and therefore determines which globally configured SiteMinder Agent Authenticator will be invoked. For example, the following login-config element would result in the SMJBossFormAuthenticator handling application requests:

   ```
   <login-config>
       <auth-method>FORM</auth-method>
       <realm-name/>
       <form-login-config>
           <form-login-page>/login.jsp</form-login-page>
           <form-error-page>/fail_login.jsp</form-error-page>
       </form-login-config>
   </login-config>
   ```

6. Save the file and exit the text editor

7. Install or update the web application.

Map Web Applications to the SiteMinderDomain Security Domain

Create a jboss-web.xml deployment descriptor file that defines the SiteMinderDomain as the security domain for *each* web application that you want to protect with the SiteMinder Agent. The jboss-web.xml file must be created in the application WEB-INF directory.

To map a web application to the SiteMinderDomain security domain

1. Navigate to the application WEB-INF directory.
2. Open a text editor.
3. Enter the following:

   ```
   <jboss-web>
     <security-domain>java:/jaas/SiteMinderDomain</security-domain>
   </jboss-web>
   ```

4. Save the file as jboss-web.xml and exit the text editor.

Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the SiteMinder Agent.

To restart the JBoss Application Server

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the SiteMinder Agent.

Chapter 7: Configure the SiteMinder Agent Security Interceptor to Protect Web Applications on JBoss 6.x

This section contains the following topics:

Configure the SiteMinder Agent Authenticator for Applications on JBoss 6.x

The SiteMinder Agent Authenticator extends the functionality of the JBossSX default authenticators with the ability to authenticate a user request that is based on an associated SiteMinder session cookie.

To configure a web application to use the SiteMinder Agent Authenticator to handle requests, create a jboss-web.xml file in the application WEB-INF directory. Configuring a jboss-web.xml file overrides the default authenticators.

Follow these steps:

1. Navigate to the application WEB-INF directory.
2. Open jboss-web.xml in a text editor.
3. Define a context element containing a valve subelement that specifies the class name of the SiteMinder Agent Authenticator which you want to handle application requests.

   To configure the application to use SMJBoss6BasicAuthenticator, type:

   ```
   <Valve
   className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBoss6BasicAuthen
   ticator"/>
   ```

To configure the application to use the SMJBoss6FormAuthenticator, type:

```
<Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBoss6FormAuthent
icator"/>
```

To configure the application to use SMJBoss6ClientCertAuthenticator, type:

```
<Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBoss6ClientCertA
uthenticator"/>
```

To configure the application to use SMJBoss6DigestAuthenticator, type:

```
<Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBoss6DigestAuthe
nticator"/>
```

To configure the application to use the SMJBoss6IdentityAsserter, type:

```
<Valve
className="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBoss6IdentityAss
erter"/>
```

4. Save the file and exit the text editor.

Make the CA SiteMinder® Agent Java Class Accessible to Your Applications

To protect your applications with CA SiteMinder®, they must be able to access the CA SiteMinder® Agent Java classes in module com.ca.siteminder.jbossagent. To make the CA SiteMinder® Agent Java classes accessible to your applications, do one of the following procedures:

- Configure the SiteMinder Agent as a Global Module (see page 85)
- Configure the SiteMinder Agent as an Application Dependency (see page 87)

Configure the SiteMinder Agent as a Global Module

Configure the CA SiteMinder® Agent as a global module by adding a new subsystem definition in the standalone.xml file.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. Add the following highlighted module name element to define the SiteMinder Agent as a global module in the "ee" web services subsystem:

```
<subsystem xmlns="urn:jboss:domain:ee:1.1">
  <global-modules>
    <module name="com.ca.siteminder.jbossagent" slot="main"/>
  </global-modules>

<spec-descriptor-property-replacement>false</spec-descriptor-property-replace
ment>

<jboss-descriptor-property-replacement>true</jboss-descriptor-property-replac
ement>
</subsystem>
```

4. Save the file and exit the text editor.

Notes:

- To configure the WSS Agent JAX-WS HTTP Handler to protect all JAX-WS web services, you must add the CA SiteMinder® Agent as a global module.
- Configuring the CA SiteMinder® Agent as a global module makes it accessible to all web applications (using the CA SiteMinder® Agent Security Interceptor) and web services (using the WSS Agent Security Interceptor).
- There is a conflict between the default JBoss and CA SiteMinder® xml-security libraries. If you configure the SiteMinder Agent as a global module, you must resolve that conflict (see page 86).

More information

Configure the WSS Agent JAX-WS HTTP Handler to Protect all JAX-WS HTTP Web Services on JBoss 6.x (see page 109)
Resolve a Conflict Between the JBoss and WSS Agent xml-security Libraries if the SiteMinder Agent is Defined as a Global Module (see page 86)
Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services on JBoss 6.x (see page 112)

Resolve a Conflict Between the JBoss and WSS Agent xml-security Libraries if the SiteMinder Agent is Defined as a Global Module

There is a conflict between the default JBoss and CA SiteMinder® XML Security libraries. If you configure the SiteMinder Agent as a global module, remove the JBoss XML Security library (org.apache.santuario.xmlsec) from the module definitions in module.xml.

Follow these steps:

1. Navigate to the following location:
   - **Windows:** *JBOSS_HOME\modules\system\layers\base\org\jboss\as\webservices\server\integration\main*
   - **UNIX:** *JBOSS_HOME*/modules/system/layers/base/org/jboss/as/webservices/server/integration/main
2. Open module.xml in a text editor.
3. Locate and comment out the following line:

   ```
   <!-- <module name="org.apache.santuario.xmlsec" export="true"/> -->
   ```

Note: For applications that depend on the default JBoss XML Security library, do one of the following procedures to enable them to access to it:

- Package the org.apache.santuario.xmlsec JAR files as a separate module from the JBoss web services module and configure it as a dependency for those applications.
- Include the org.apache.santuario.xmlsec JAR files in the application WAR file.

Configure the SiteMinder Agent as a Per-Application Dependency

If the CA SiteMinder® Agent is not defined as a global module, define it as a dependency in the jboss-deployment-structure.xml file of each application that you want to protect.

Follow these steps:

1. Navigate to the application WEB-INF directory.
2. Open jboss-deployment-structure.xml in a text editor.
3. Add the following module name element to the dependencies element:

   ```
   <module name="com.ca.siteminder.jbossagent" />
   ```

   For example:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <jboss-deployment-structure>
      <deployment>
          <dependencies>
                <module name="com.ca.siteminder.jbossagent" />
          </dependencies>
      </deployment>
   </jboss-deployment-structure>
   ```

4. Save the file and exit the text editor.

Define a JBossSX Security Domain for the SiteMinder Agent Login Module on JBoss 6.x

Define a JBoss security domain named SiteMinderDomain that configures the SiteMinder Agent Login Module required to authenticate credentials obtained by SiteMinder Agent authenticators. Configure the SiteMinderDomain by adding a <security-domain-name> element to the standalone.xml file.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows:** *JBOSS_HOME\standalone\configuration*
   - **UNIX:** *JBOSS_HOME*/standalone/configuration
2. Open the standalone.xml file in a text editor.

3. Add the following <security-domain-name> element:

```
<security-domain name="SiteMinderDomain" cache-type="default">
    <authentication>
            <login-module
code="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossLoginModule"
flag="required"/>
    </authentication>
</security-domain>
```

4. Save the file and exit the text editor.

Configure Web Applications to Invoke the SiteMinder Agent Security Interceptor

To protect a web application using the SiteMinder Agent Security Interceptor, edit its deployment descriptor to enable security and map it to the SiteMinderDomain security domain.

Edit the Application Deployment Descriptor to Enable Security

Edit the web.xml deployment descriptor to enable security for each web application that you want to protect with the SiteMinder Agent Web Interceptor. The web.xml file is located in the application WEB-INF directory.

For more information about the web.xml file and constituent element syntax, see the JBoss Enterprise Application Platform documentation.

To Edit the web.xml deployment descriptor to enable security

1. Navigate to the web application WEB-INF directory
2. Open the web.xml deployment descriptor file in a text editor.

3. Add one or more security-constraint elements defining what resources in the web application are to be protected. For example:

   ```
   <security-constraint>
       <display-name>Constraint1</display-name>
       <web-resource-collection>
           <web-resource-name>admin resource</web-resource-name>
           <description/>
           <url-pattern>/admin/*</url-pattern>
           <http-method>GET</http-method>
           <http-method>POST</http-method>
       </web-resource-collection>
       <auth-constraint>
           <description/>
           <role-name>adminRole</role-name>
       </auth-constraint>
   </security-constraint>
   ```

4. Add a security-role element defining roles used by the application. For example:

   ```
   <security-role>
        <description/>
        <role-name>adminRole</role-name>
   </security-role>
   ```

5. Add a login-config element. The auth-method subelement of the login-config element defines the authentication method (BASIC, FORMS, and so on) and therefore determines which globally configured SiteMinder Agent Authenticator will be invoked. For example, the following login-config element would result in the SMJBossFormAuthenticator handling application requests:

   ```
   <login-config>
       <auth-method>FORM</auth-method>
       <realm-name/>
       <form-login-config>
           <form-login-page>/login.jsp</form-login-page>
           <form-error-page>/fail_login.jsp</form-error-page>
       </form-login-config>
   </login-config>
   ```

6. Save the file and exit the text editor

7. Install or update the web application.

Map Web Applications to the SiteMinderDomain Security Domain

Create a jboss-web.xml deployment descriptor file that defines the SiteMinderDomain as the security domain for *each* web application that you want to protect with the SiteMinder Agent. The jboss-web.xml file must be created in the application WEB-INF directory.

To map a web application to the SiteMinderDomain security domain

1. Navigate to the application WEB-INF directory.
2. Open a text editor.
3. Enter the following:

```
<jboss-web>
  <security-domain>java:/jaas/SiteMinderDomain</security-domain>
</jboss-web>
```

4. Save the file as jboss-web.xml and exit the text editor.

Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the SiteMinder Agent.

To restart the JBoss Application Server

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the SiteMinder Agent.

Chapter 8: Configure SiteMinder Policies to Protect Web Applications

This section contains the following topics:

Configure a SiteMinder Agent Security Interceptor Authentication Realm

Configure a realm on the Policy Server to allow the SiteMinder Agent Security Interceptor to validate users' credentials using information obtained from SiteMinder session cookies. You use the SiteMinder Administrative UI to create the SiteMinder Agent Security Interceptor authentication realm.

For more information about the SiteMinder Administrative UI and its use to create domains and realms, see the *CA SiteMinder® Policy Server Configuration Guide*.

Follow these steps:

1. Click Policies, Domains.
2. Click Domain, Create Domain.
3. The Create Domain pane opens.

   Note: You can click Help for a description of fields, controls, and their respective requirements.

4. Type the name and a description of the Domain in the fields on the General group box.
5. Add one or more user directories that contain the users who can access the protected resources.

6. Create the validation realm:

   a. Click the Realms tab on the Domain pane, New Realm, OK.

   b. The Create Realm pane opens.

   c. Enter the following information:

      - Name: A unique name for the realm—for example, SiteMinder Agent Security Interceptor Validation Realm

      - Description: An optional description for the validation realm

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

      - Resource Filter: /smauthenticationrealm

      - Authentication Scheme: Basic

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

   d. Specify session properties in the Session group box:

      - Disable all session time-outs

      - Ensure the No Persistent Session option is selected

   e. Click Finish.

      The Create Realm Task is submitted for processing.

7. Click Submit.

   The Create Domain Task is submitted for processing.

(Optional) Configure the Agent to Return Group Membership to JBoss Using Responses

The SiteMinder Agent Web Interceptor can be configured to return physical or virtual group membership information to JBoss using SiteMinder HTTP header responses from the Policy Server during user authentication.

When the SiteMinder Agent Web Interceptor receives responses containing the _SM_JBOSS_GROUP=*group name* syntax, the SiteMinder Agent Web Interceptor converts the *group_name* value to a J2EE principal and adds this principal to the subject after successful authentication.

group_name

> Specifies 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 adds the same amount of group principals as responses received from the Policy Server.

Note: The SiteMinder Agent Web Interceptor can only process _SM_JBOSS_GROUP response attributes to return group membership information to JBoss. It cannot process other response attributes added to HTTP header variables to pass information to a web application.

To configure Groups as responses for the SiteMinder Agent

1. Configure an OnAuthAccept group authentication rule with a * resource filter in the SiteMinder Authentication Realm.
2. Create SiteMinder HTTP header responses using the _SM_JBOSS_GROUP variable name in the policy domain for the SiteMinder Authentication Realm.

   Note: The SiteMinder Administrative UI shows an additional underscore before "_SM_JBOSS_GROUP" when it displays the variable name, so that it appears as "HTTP__SM_JBOSS_GROUP". This is not an error and can be ignored.
3. In the policy domain for the SiteMinder Authentication Realm:

   a. Create a group policy.

   b. Attach the users who belong to the group policy.

   c. Attach the group authentication rule to this policy.

   d. Bind the group response to the group authentication rule.

The following example shows one method of configuring the SiteMinder Agent Web Interceptor to return groups using 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 JBoss groups:

Name	Attribute Kind	Variable Name	Variable Value
Group Administrators	Static HTTP Header	_SM_JBOSS_GROUP	Administrators
Group Deployers	Static HTTP Header	_SM_JBOSS_GROUP	Deployers
Group Monitors	Static HTTP Header	_SM_JBOSS_GROUP	Monitors
Group Operators	Static HTTP Header	_SM_JBOSS_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.
4. Repeat Step 3 to configure separate policies for the Deployers, Operators, and Monitors groups.

Configure Security Policies for the Proxy Server Web Agent

To configure the SiteMinder Agent for JBoss to protect web applications by perimeter authentication, create policies that specify how the Web Agent on the proxy server controls access to the URL that represents the proxied JBoss web application resources.

Note: For complete information about SiteMinder policy configuration for web resources, see the *CA SiteMinder Policy Configuration Guide*.

Chapter 9: Configure the WSS Agent Security Interceptor to Protect Web Services on JBoss 5.x

This section contains the following topics:

Configure WSS Agent Security Interceptor Protection for JAX-RPC Web Services Over HTTP Transport

To configure the WSS Agent Web Interceptor to protect JAX-RPC web services over HTTP transport, configure those services to invoke the WSS Agent JAX-RPC HTTP Handler. You can configure global use of the JAX-RPC Handler for all JAX-RPC HTTP web services or configure it for individual web services, as required.

Configure the WSS Agent JAX-RPC HTTP Handler for all JAX-RPC HTTP Web Services

To configure the WSS Agent JAX-RPC Handler to be invoked for all JAX-RPC HTTP web services, add the WSS Agent JAX-RPC Handler class (com.ca.soa.agent.jaxrpcplugin.JaxrpcHandler) to the standard JAX-RPC endpoint configuration file, standard-jaxrpc-endpoint-config.xml.

By default, the standard-jaxrpc-endpoint-config.xml file is in the following location:

JBOSS_HOME/server/*instance_type*/deployers/jbossws.deployer/META-INF

instance_type

Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

Follow these steps:

1. Navigate to the location of the standard-jaxrpc-endpoint-config.xml file for your JBoss version and instance type.
2. Open the standard-jaxrpc-endpoint-config.xml file in a text editor.
3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined.

   ```
   <handler>
      <j2ee:handler-name>SM XMLAgentJaxrpc Handler</j2ee:handler-name>
      <j2ee:handler-class>
        com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler
      </j2ee:handler-class>
   </handler>
   ```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-RPC Handler will be invoked for all JAX-RPC web services.

Example standard-jaxrpc-endpoint-config.xml file

```
<jaxrpc-config xmlns="urn:jboss:jaxrpc-config:2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:j2ee="http://java.sun.com/xml/ns/j2ee"
xsi:schemaLocation="urn:jboss:jaxrpc-config:2.0 jaxrpc-config_2_0.xsd">

    <endpoint-config>
      <config-name>Standard Endpoint</config-name>
        <pre-handler-chain>
          <handler-chain-name>SM XMLAgentJaxrpc Handlers</handler-chain-name>
          <handler>
            <j2ee:handler-name>SM XMLAgentJaxrpc Handler</j2ee:handler-name>
            <j2ee:handler-class>
              com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler
            </j2ee:handler-class>
          </handler>
        </pre-handler-chain>
  </endpoint-config>

</jaxrpc-config>
```

Configure the WSS Agent JAX-RPC HTTP Handler for a Single Web Service

Configure individual JAX-RPC HTTP web services to invoke the WSS Agent JAX-RPC HTTP Handler by defining the com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler in the application webservices.xml deployment descriptor.

For example:

```
<webservices ...>
   <webservice-description>
    ...
    <port-component>
      ...
      <handler>
        <handler-name>SM XMLAgentJaxrpc Handler</handler-name>

<handler-class>com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler</handler-c
lass>
      </handler>
    </port-component>
   </webservice-description>
  </webservices>
```

The JBoss WSS Agent JAX-RPC HTTP Handler will be invoked only for this web service.

Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over HTTP Transport

To configure the WSS Agent Security Interceptor to protect JAX-WS web services over HTTP transport, configure those services to invoke the WSS Agent JAX-WS HTTP Handler. You can configure global use of the JAX-WS Handler for all JAX-WS HTTP web services or configure it for individual web services, as required.

Configure the WSS Agent JAX-WS HTTP Handler for all JAX-WS HTTP Web Services

To configure the WSS Agent JAX-WS HTTP Handler to be invoked for all JAX-WS HTTP web services, add the WSS Agent JAX-WS Handler class (com.ca.soa.agent.jaxwsplugin.JaxWsHandler) to the standard JAX-WS endpoint configuration file, standard-jaxws-endpoint-config.xml.

By default, the standard-jaxws-endpoint-config.xml file is in the following location:

JBOSS_HOME/server/*instance_type*/deployers/jbossws.deployer/META-INF

instance_type

Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

Follow these steps:

1. Navigate to the location of the standard-jaxws-endpoint-config.xml file for your JBoss version and instance type.
2. Open the standard-jaxws-endpoint-config.xml file in a text editor.
3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined:

   ```
   <javaee:handler>
      <javaee:handler-name>
         JBoss JAX-WS PEP Interceptor
      </javaee:handler-name>
      <javaee:handler-class>
         com.ca.soa.agent.jaxwsplugin.JaxWsHandler
      </javaee:handler-class>
   </javaee:handler>
   ```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler will be invoked for all JAX-WS web services.

Example standard-jaxws-endpoint-config.xml file

```
<jaxws-config xmlns="urn:jboss:jaxws-config:2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:javaee="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="urn:jboss:jaxws-config:2.0 schema/jaxws-config_2_0.xsd">

<endpoint-config>
    <config-name>Standard Endpoint</config-name>
    <pre-handler-chains>
      <javaee:handler-chain>
        <javaee:protocol-bindings>##SOAP11_HTTP</javaee:protocol-bindings>

        <javaee:handler>
           <javaee:handler-name>
                JBoss JAX-WS PEP Interceptor
           </javaee:handler-name>
           <javaee:handler-class>
                com.ca.soa.agent.jaxwsplugin.JaxWsHandler
           </javaee:handler-class>
        </javaee:handler>
```

```
        <javaee:handler>
          <javaee:handler-name>Recording Handler</javaee:handler-name>
          <javaee:handler-class>
                org.jboss.wsf.framework.invocation.RecordingServerHandler
          </javaee:handler-class>
        </javaee:handler>

      </javaee:handler-chain>
    </pre-handler-chains>
</endpoint-config>
```

Configure the WSS Agent JAX-WS HTTP Handler for a Single JAX-WS HTTP Web Service

Configure individual JAX-WS HTTP web services to invoke the WSS Agent JAX-WS Handler.

Follow these steps:

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

```
<?xml version="1.0" encoding="UTF-8"?>
<handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
  <handler-chain>
    <handler>
      <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>

<handler-class>com.ca.soa.agent.jaxwsplugin.JaxWsHandler</handler-class>
    </handler>
  </handler-chain>
</handler-chains>
```

2. Add the following JWS annotation to the web service JWS file:

```
@HandlerChain(file = "Services_handler.xml")
```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service (see page 84).

The JBoss WSS Agent JAX-WS Handler is invoked for the web service.

Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over JMS Transport

To configure the WSS Agent Web Interceptor to protect JAX-WS web services over JMS transport, configure those services to invoke the WSS Agent JAX-WS JMS Handler. You can configure global use of the JAX-WS JMS Handler for all JAX-WS JMS web services or configure it for individual web services, as required.

Important! Do not place the WSS Agent JAX-WS HTTP Handler and the WSS Agent JAX-WS JMS Handler in the same handler chain. If you configure either handler in the default handler chain for the container, verify that all JAX-WS web services in the container use the corresponding transport.

Important! Do not place the WSS Agent JAX-WS HTTP Handler and the WSS Agent JAX-WS JMS Handler in the same handler chain. If you configure either handler in the default handler chain for the container, verify that all JAX-WS web services in the container use the corresponding transport.

Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services

To configure the WSS Agent JAX-WS JMS Handler to be invoked for all JAX-WS JMS web services, add the WSS Agent JAX-WS JMS Handler class (com.ca.soa.agent.jmsplugin.JaxWsJMSHandler) to the standard JAX-WS endpoint configuration file, standard-jaxws-endpoint-config.xml.

The standard-jaxws-endpoint-config.xml file is located in *JBOSS_HOME*/server/*instance_type*/deployers/jbossws.deployer/META-INF.

instance_type

Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

Follow these steps:

1. Navigate to *JBOSS_HOME*/server/default/deployers/jbossws.deployer/META-INF.
2. Open the standard-jaxws-endpoint-config.xml file in a text editor.

3. Add the following javaee:handler element to the "Standard Endpoint" endpoint-config element as the first such element defined:

   ```
   <javaee:handler>
      <javaee:handler-name>
         JBoss JAX-WS PEP Interceptor
      </javaee:handler-name>
      <javaee:handler-class>
         com.ca.soa.agent.jmsplugin.JaxWsJMSHandler
      </javaee:handler-class>
   </javaee:handler>
   ```

4. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler will be invoked for all JAX-WS web services.

Example standard-jaxws-endpoint-config.xml file

```
<jaxws-config xmlns="urn:jboss:jaxws-config:2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:javaee="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="urn:jboss:jaxws-config:2.0 schema/jaxws-config_2_0.xsd">

<endpoint-config>
    <config-name>Standard Endpoint</config-name>
    <pre-handler-chains>
      <javaee:handler-chain>
        <javaee:protocol-bindings>##SOAP11_HTTP</javaee:protocol-bindings>

        <javaee:handler>
           <javaee:handler-name>
                JBoss JAX-WS PEP Interceptor
           </javaee:handler-name>
           <javaee:handler-class>
                com.ca.soa.agent.jmsplugin.JaxWsJMSHandler
           </javaee:handler-class>
        </javaee:handler>

        <javaee:handler>
          <javaee:handler-name>Recording Handler</javaee:handler-name>
          <javaee:handler-class>
                org.jboss.wsf.framework.invocation.RecordingServerHandler
          </javaee:handler-class>
        </javaee:handler>

      </javaee:handler-chain>
    </pre-handler-chains>
</endpoint-config>
```

Configure the WSS Agent JAX-WS Handler for a Single JAX-WS JMS Web Service

You can configure individual JAX-WS JMS web services to invoke the WSS Agent JAX-WS JMS Handler.

Follow these steps:

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
     <handler-chain>
       <handler>
         <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>

   <handler-class>com.ca.soa.agent.jmsplugin.JaxWsJMSHandler</handler-class>
       </handler>
     </handler-chain>
   </handler-chains>
   ```

2. Add the following JWS annotation to the web service JWS file:

   ```
   @HandlerChain(file = "Services_handler.xml")
   ```

The JBoss WSS Agent JAX-WS Handler will be invoked only for this web service.

Configure the WSS Agent Login Module

Define a JBoss security domain named system.XMLAgent that configures the WSS Agent Login Module required to authenticate credentials obtained by the WSS Agent Handlers.

You configure the system.XMLAgent by adding an application-policy element to the login-config.xml file located in *JBOSS_HOME*/server/*instance_type*/conf.

instance_type

Specifies the JBoss Application Server instance type (one of default, minimal, production, standard, or web).

To configure SiteMinder Agent Authenticators at the global level

1. Navigate to server/*server_name*/conf/
2. Open the login-config.xml file in a text editor.
3. Add the following application-policy element defining the SiteMinderDomain:

   ```
   <application-policy name="system.XMLAgent">
       <authentication>
         <login-module code="com.ca.soa.agent.appserver.jaas.XMLAgentLoginModule"
           flag="required">
   ```

```
        <module-option
name="unauthenticatedIdentity">anonymous</module-option>
      </login-module>
    </authentication>
</application-policy>
```

4. Save the file and exit the text editor.

Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the SiteMinder Agent.

To restart the JBoss Application Server

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the SiteMinder Agent.

Chapter 10: Configure the WSS Agent Security Interceptor to Protect Web Services on JBoss 6.x

This section contains the following topics:

Make the CA SiteMinder® Agent Java Class Accessible to Your Applications

To protect your applications with CA SiteMinder®, they must be able to access the CA SiteMinder® Agent Java classes in module com.ca.siteminder.jbossagent. To make the CA SiteMinder® Agent Java classes accessible to your applications, do one of the following procedures:

- Configure the SiteMinder Agent as a Global Module (see page 85)
- Configure the SiteMinder Agent as an Application Dependency (see page 87)

Configure the SiteMinder Agent as a Global Module

Configure the CA SiteMinder® Agent as a global module by adding a new subsystem definition in the standalone.xml file.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. Add the following highlighted module name element to define the SiteMinder Agent as a global module in the "ee" web services subsystem:

   ```
   <subsystem xmlns="urn:jboss:domain:ee:1.1">
     <global-modules>
       <module name="com.ca.siteminder.jbossagent" slot="main"/>
     </global-modules>

   <spec-descriptor-property-replacement>false</spec-descriptor-property-replace
   ment>

   <jboss-descriptor-property-replacement>true</jboss-descriptor-property-replac
   ement>
   </subsystem>
   ```

4. Save the file and exit the text editor.

Notes:

- To configure the WSS Agent JAX-WS HTTP Handler to protect all JAX-WS web services, you must add the CA SiteMinder® Agent as a global module.
- Configuring the CA SiteMinder® Agent as a global module makes it accessible to all web applications (using the CA SiteMinder® Agent Security Interceptor) and web services (using the WSS Agent Security Interceptor).
- There is a conflict between the default JBoss and CA SiteMinder® xml-security libraries. If you configure the SiteMinder Agent as a global module, you must resolve that conflict (see page 86).

More information

Configure the WSS Agent JAX-WS HTTP Handler to Protect all JAX-WS HTTP Web Services on JBoss 6.x (see page 109)
Resolve a Conflict Between the JBoss and WSS Agent xml-security Libraries if the SiteMinder Agent is Defined as a Global Module (see page 86)
Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services on JBoss 6.x (see page 112)

Resolve a Conflict Between the JBoss and WSS Agent xml-security Libraries if the SiteMinder Agent is Defined as a Global Module

There is a conflict between the default JBoss and CA SiteMinder® XML Security libraries. If you configure the SiteMinder Agent as a global module, remove the JBoss XML Security library (org.apache.santuario.xmlsec) from the module definitions in module.xml.

Follow these steps:

1. Navigate to the following location:
   - **Windows:**
     JBOSS_HOME\modules\system\layers\base\org\jboss\as\webservices\server\integration\main
   - **UNIX:**
     JBOSS_HOME/modules/system/layers/base/org/jboss/as/webservices/server/integration/main
2. Open module.xml in a text editor.
3. Locate and comment out the following line:

   ```
   <!-- <module name="org.apache.santuario.xmlsec" export="true"/> -->
   ```

Note: For applications that depend on the default JBoss XML Security library, do one of the following procedures to enable them to access to it:

- Package the org.apache.santuario.xmlsec JAR files as a separate module from the JBoss web services module and configure it as a dependency for those applications.
- Include the org.apache.santuario.xmlsec JAR files in the application WAR file.

Configure the SiteMinder Agent as a Per-Application Dependency

If the CA SiteMinder® Agent is not defined as a global module, define it as a dependency in the jboss-deployment-structure.xml file of each application that you want to protect.

Follow these steps:

1. Navigate to the application WEB-INF directory.
2. Open jboss-deployment-structure.xml in a text editor.
3. Add the following module name element to the dependencies element:

   ```
   <module name="com.ca.siteminder.jbossagent" />
   ```

   For example:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <jboss-deployment-structure>
      <deployment>
          <dependencies>
                <module name="com.ca.siteminder.jbossagent" />
          </dependencies>
      </deployment>
   </jboss-deployment-structure>
   ```

4. Save the file and exit the text editor.

Configure the WSS Agent JAX-RPC HTTP Handler to Protect Web Services in JBoss 6.x

Configure each JAX-RPC HTTP web service to invoke the WSS Agent JAX-RPC HTTP Handler.

Note: There is no global way to configure the WSS Agent JAX-RPC HTTP Handler to protect all JAX_RPC web services.

Follow these steps:

1. Open the application webservices.xml deployment descriptor in a text editor.
2. Define the com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler.

For example:

```
<webservices ...>
   <webservice-description>
    ...
    <port-component>
      ...
      <handler>
        <handler-name>SM XMLAgentJaxrpc Handler</handler-name>

<handler-class>com.ca.soa.agent.appserver.jaxrpc.XMLAgentJaxrpcHandler</handl
er-class>
      </handler>
    </port-component>
   </webservice-description>
  </webservices>
```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service (see page 84).

The JBoss WSS Agent JAX-RPC HTTP Handler is invoked only for this web service.

Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over HTTP Transport

To configure the WSS Agent Security Interceptor to protect JAX-WS web services over HTTP transport, configure those services to invoke the WSS Agent JAX-WS HTTP Handler. You can configure global use of the JAX-WS Handler for all JAX-WS HTTP web services or configure it for individual web services, as required.

Configure the WSS Agent JAX-WS HTTP Handler to Protect all JAX-WS HTTP Web Services on JBoss 6.x

To configure the WSS Agent Security Interceptor to protect all JAX-WS HTTP web services, make the following changes to standalone.xml:

- Add a subsystem definition to configure the agent as a global module (if it is not already present).
- Add a pre-handler-chain definition to configure the WSS Agent JAX-WS HTTP Handler as the handler for all JAX-WS HTTP web services.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone-full.xml in a text editor.
3. If it is not already defined, configure the SiteMinder Agent as a global module (see page 85)
4. Add the following pre-handler-chain element to the "Standard Endpoint" endpoint-config element in the web services subsystem definition as the first such element defined:

   ```
   <pre-handler-chain name="WSSAgent" protocol-bindings="##SOAP11_HTTP
   ##SOAP12_HTTP">
           <handler name="SoaJaxWsHandler"
   class="com.ca.soa.agent.jaxwsplugin.JaxWsHandler"/>
    </pre-handler-chain>
   ```

   Note: The default standalone-full.xml does not have a web services subsystem predefined. If no web services subsystem is present, add one that includes the previous pre-handler-chain element. For example:

   ```
   <subsystem xmlns="urn:jboss:4domain:webservices:1.2">
     <modify-wsdl-address>true</modify-wsdl-address>
     <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
     <endpoint-config name="Standard-Endpoint-Config">
        <pre-handler-chain name="WSSAgent" protocol-bindings="##SOAP11_HTTP
   ##SOAP12_HTTP">
           <handler name="SoaJaxWsHandler"
   class="com.ca.soa.agent.jaxwsplugin.JaxWsHandler"/>
        </pre-handler-chain>
     </endpoint-config>
     <client-config name="Standard-Client-Config"/>
   </subsystem>
   ```

5. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler is invoked for all JAX-WS HTTP web services.

Configure the WSS Agent JAX-WS HTTP Handler for a Single JAX-WS HTTP Web Service

Configure individual JAX-WS HTTP web services to invoke the WSS Agent JAX-WS Handler.

Follow these steps:

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
     <handler-chain>
       <handler>
         <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>

   <handler-class>com.ca.soa.agent.jaxwsplugin.JaxWsHandler</handler-class>
       </handler>
     </handler-chain>
   </handler-chains>
   ```

2. Add the following JWS annotation to the web service JWS file:

   ```
   @HandlerChain(file = "Services_handler.xml")
   ```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service (see page 84).

The JBoss WSS Agent JAX-WS Handler is invoked for the web service.

Configure WSS Agent Security Interceptor Protection for JAX-WS Web Services Over JMS Transport on JBoss 6.x

To configure the WSS Agent Web Interceptor to protect JAX-WS web services over JMS transport, configure those services to invoke the WSS Agent JAX-WS JMS Handler. You can configure global use of the JAX-WS JMS Handler for all JAX-WS JMS web services or configure it for individual web services, as required.

Important! Do not place the WSS Agent JAX-WS HTTP Handler and the WSS Agent JAX-WS JMS Handler in the same handler chain. If you configure either handler in the default handler chain for the container, verify that all JAX-WS web services in the container use the corresponding transport.

Configure the WSS Agent JAX-WS JMS Handler for all JAX-WS JMS Web Services on JBoss 6.x

To configure the WSS Agent Security Interceptor to protect all JAX-WS JMS web services, make the following changes to standalone.xml:

- Add a subsystem definition to configure the agent as a global module (if it is not already present).
- Add a pre-handler-chain definition to configure the WSS Agent JAX-WS JMS Handler as the handler for all JAX-WS JMS web services.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows**: *JBOSS_HOME*\standalone\configuration
   - **UNIX**: *JBOSS_HOME*/standalone/configuration
2. Open standalone.xml in a text editor.
3. If it is not already defined, configure the SiteMinder Agent as a global module (see page 85)
4. Add the following pre-handler-chain element to the "Standard Endpoint" endpoint-config element in the web services subsystem definition as the first such element defined:

```
<pre-handler-chain name="WSSAgent" protocol-bindings="##SOAP11_HTTP
##SOAP12_HTTP">
        <handler name="SoaJaxWsJMSHandler"
class="com.ca.soa.agent.jaxwsplugin.JaxWsJMSHandler"/>
 </pre-handler-chain>
```

   Note: The default standalone.xml does not have a web services subsystem predefined. If no web services subsystem is present, add one that includes the previous pre-handler-chain element. For example:

```
<subsystem xmlns="urn:jboss:4domain:webservices:1.2">
  <modify-wsdl-address>true</modify-wsdl-address>
  <wsdl-host>${jboss.bind.address:127.0.0.1}</wsdl-host>
  <endpoint-config name="Standard-Endpoint-Config">
     <pre-handler-chain name="WSSAgent" protocol-bindings="##SOAP11_HTTP
##SOAP12_HTTP">
        <handler name="SoaJaxWsJMSHandler"
class="com.ca.soa.agent.jaxwsplugin.JaxWsJMSHandler"/>
     </pre-handler-chain>
  </endpoint-config>
  <client-config name="Standard-Client-Config"/>
</subsystem>
```

5. Save the file and exit the text editor.

The JBoss WSS Agent JAX-WS Handler is invoked for all JAX-WS JMS web services.

Configure the WSS Agent JAX-WS Handler for a Single JAX-WS JMS Web Service on JBoss 6.x

You can configure individual JAX-WS JMS web services to invoke the WSS Agent JAX-WS JMS Handler.

Follow these steps:

1. Create a handler chain configuration file, for example, Services_handler.xml, containing the following text:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <handler-chains xmlns="http://java.sun.com/xml/ns/javaee">
     <handler-chain>
       <handler>
         <handler-name>JBoss JAX-WS PEP Interceptor</handler-name>

   <handler-class>com.ca.soa.agent.jmsplugin.JaxWsJMSHandler</handler-class>
       </handler>
     </handler-chain>
   </handler-chains>
   ```

2. Add the following JWS annotation to the web service JWS file:

   ```
   @HandlerChain(file = "Services_handler.xml")
   ```

3. Verify that the CA SiteMinder® Agent Java class is accessible to the web service (see page 84).

The JBoss WSS Agent JAX-WS Handler is invoked only for this web service.

Define a JBossSX Security Domain for the SiteMinder Agent Login Module on JBoss 6.x

Define a JBoss security domain named SiteMinderDomain that configures the SiteMinder Agent Login Module required to authenticate credentials obtained by SiteMinder Agent authenticators. Configure the SiteMinderDomain by adding a <security-domain-name> element to the standalone.xml file.

Follow these steps:

1. Navigate to one of the following locations:
   - **Windows:** *JBOSS_HOME\standalone\configuration*
   - **UNIX:** *JBOSS_HOME*/standalone/configuration
2. Open the standalone.xml file in a text editor.

3. Add the following <security-domain-name> element:

```
<security-domain name="SiteMinderDomain" cache-type="default">
    <authentication>
            <login-module
code="com.ca.soa.agent.appserver.authenticator.jBoss.SMJBossLoginModule"
flag="required"/>
    </authentication>
</security-domain>
```

4. Save the file and exit the text editor.

Restart the JBoss Application Server

Restart the JBoss Application Server to commit configuration changes you made for the SiteMinder Agent.

To restart the JBoss Application Server

1. If necessary, stop the JBoss Application Server process.
2. Open a command window.
3. Navigate to the *JBOSS_HOME*/bin directory.
4. Run the run.bat (Windows) or run.sh (UNIX) script.

The JBoss Application Server restarts with the configuration changes you made for the SiteMinder Agent.

Chapter 11: Troubleshooting

Messages Are Not in the Expected Format Upon WSS Agent JAX-RPC Handler Authentication Failure

Symptom:

Upon an authentication failure, messages returned by the WSS Agent JAX-RPC Handler do not result in the expected "Authentication Failure" message. Instead, JBoss returns the following message:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
   <env:Header/>
   <env:Body>
      <env:Fault>
         <faultcode>env:Server</faultcode>
         <faultstring>JBWS025230: Illegal faultcode
'{http://schemas.xmlsoap.org/soap/envelope/}Client.Authentication', allowed values
are: [{http://schemas.xmlsoap.org/soap/envelope/}Client,
{http://schemas.xmlsoap.org/soap/envelope/}Server,
{http://schemas.xmlsoap.org/soap/envelope/}VersionMismatch,
{http://schemas.xmlsoap.org/soap/envelope/}MustUnderstand]</faultstring>
      </env:Fault>
   </env:Body>
</env:Envelope>
```

Solution:

By default, the JAX-RPC Handler provides extensible SOAPFault codes. However, some versions of JBoss can only handle simple SOAPFault codes, resulting in the previous message instead of the expected "Authentication Failure" message. If your version of JBoss does not handle extensible fault codes, configure the WSS Agent JAX-RPC Handler to provide simple fault codes.

Important! Configuring this property breaks SOAPFault on JAX-WS.

Follow these steps:

1. Navigate to the following location:
   - **Windows**: *JBOSS_HOME*\bin
   - **UNIX**: *JBOSS_HOME*/bin
2. Open the following file in a text editor:
   - **Windows**: standalone.conf.bat
   - **UNIX**: standalone.conf
3. Add the following line:

   ```
   javax.xml.soap.SOAPFactory=org.jboss.ws.core.soap.SOAPFactoryImpl
   ```
4. Save your changes.
5. To apply the changes, restart the JBoss Application Server.

WSS Agent Fails to Generate Signed SAML Session Ticket Responses

Symptom:

The WSS Agent fails to generate signed SAML Session Ticket responses, producing the following error message:

```
[ERROR] stack.jbws.RequestHandlerImpl 8A2AF0AB-705F-08EF-DD11-2AA1C4AADF50 - Error
processing web service request
org.jboss.ws.WSException: ava.lang.ArrayIndexOutOfBoundsException
```

This error is because the generated SAML Session Ticket response results in an HTTP header larger than the JBoss default size limit of 4096.

Solution:

Increase the value of the maxHttpHeaderSize parameter in the JBoss server.xml file from 4096 to a value large enough to accommodate the generated header (for example, 8192).

The server.xml file is located in *JBOSS_HOME*\server\default\deploy\jbossweb.sar.

Note: For JBoss 5.1.2, increase the value of the maxHttpHeaderSize parameter in the HTTP/1.1 Connector and the AJP 1.3 Connector sections. For example:

```
<!-- A HTTP/1.1 Connector on port 8080 -->
<Connector protocol="HTTP/1.1" port="8080" address="${jboss.bind.address}"
connectionTimeout="20000" redirectPort="8443" maxHttpHeaderSize="32768"/>

<!-- A AJP 1.3 Connector on port 8009 -->
<Connector protocol="AJP/1.3" port="8009" address="${jboss.bind.address}"
redirectPort="8443" maxHttpHeaderSize="32768" />
```