integrity™ lifecycle manager

Server for CA SiteMinder®

11.1

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

1

Introduction

This guide is a supplement to the *Integrity Lifecycle Manager Installation and Upgrading Guide*.

With this release of Integrity Lifecycle Manager, PTC® provides support for user authentication using CA SiteMinder® (SiteMinder) 6.0 or 12.5.

The functionality allows Integrity Lifecycle Manager client systems to connect to the Integrity Lifecycle Manager server using a SiteMinder session token, rather than the user’s credentials. The server then validates the session token with SiteMinder, only allowing the client connection based on a successful validation.

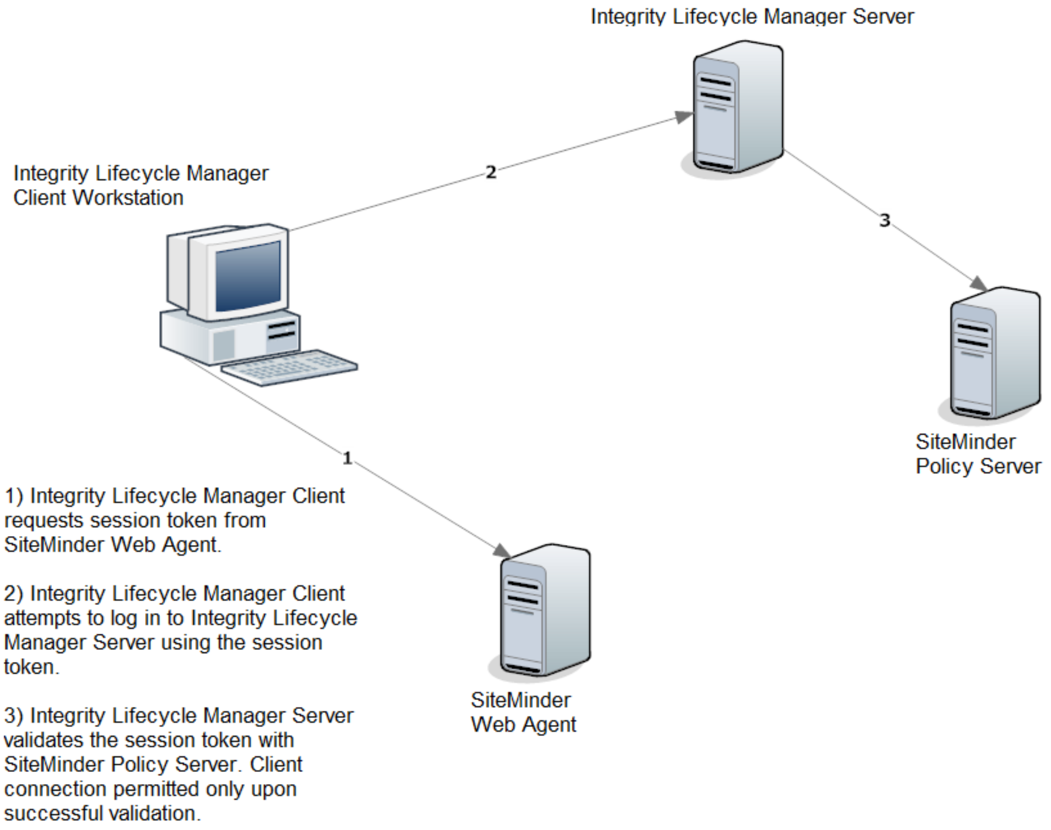
SiteMinder is a Web-based, security management system that supports centralized authentication and security policy enforcement. SiteMinder also provides single sign-on support for Web-based applications, and further allows API and Web services to integrate authentication and single signon support for non-Web applications.

This document provides details on configuring policy settings for the SiteMinder Policy Server and Web Agent, setting the required security properties on the Integrity Lifecycle Manager server, and enabling secure communications. Once the configuration steps are completed, the server is ready to authenticate users through SiteMinder.

Configuration Overview

Authentication through the SiteMinder domain is supported for the Integrity Lifecycle Manager client graphical user interface (GUI), Web interface, command line interface (CLI), and for the Integrity Lifecycle Manager API using a client-side integration point. The implementation supports the MKS Domain, as well as all other supported domains for Integrity Lifecycle Manager.

The following illustrates the general network configuration for integrating Integrity Lifecycle Manager into a SiteMinder domain:


Integrity Lifecycle Manager Server
Integrity Lifecycle Manager Client Workstation
2
3
1
SiteMinder Policy Server
SiteMinder Web Agent

1) Integrity Lifecycle Manager Client requests session token from SiteMinder Web Agent.

2) Integrity Lifecycle Manager Client attempts to log in to Integrity Lifecycle Manager Server using the session token.

3) Integrity Lifecycle Manager Server validates the session token with SiteMinder Policy Server. Client connection permitted only upon successful validation.

Network Configuration: Integrity Lifecycle Manager and SiteMinder

Alternatively, you can also configure direct communications between SiteMinder and the Integrity Lifecycle Manager server, allowing users to connect to the server without first going through a Web Agent. For more information, see Security Domains on page 18.

To determine which configuration is appropriate for your environment, contact PTC Technical Support (http://www.support ptc.com/integrity.htm) for more information.

To start using SiteMinder as the authentication point, you must configure the SiteMinder Policy Server, as well as certain security properties on the Integrity Lifecycle Manager server. The required steps are:

Configuration Step	See
Set up the Integrity Lifecycle Manager Agent to communicate with the SiteMinder Policy Server.	Configuring the SiteMinder Policy Server on page 11
Install the Integrity Lifecycle Manager server, which includes updates to support the integration with SiteMinder.	Installing the Integrity Lifecycle Manager Server on page 15
Set up the required security properties for the Integrity Lifecycle Manager server to communicate with SiteMinder, including configuration of Basic or Form authentication.	Configuring the Integrity Lifecycle Manager Properties for SiteMinder on page 17
Set up any Integrity Lifecycle Manager proxies to work with SiteMinder, and understand the required connections settings for Integrity Lifecycle Manager client systems connecting through a proxy.	Configuring FSA for SiteMinder on page 29
Configure the Integrity Lifecycle Manager server for Secure Sockets Layer (SSL) communications.	Enabling SSL Communications on page 33
Check your configuration.	Diagnostics on page 37

Assumptions

This document assumes that SiteMinder is already installed at your site and that you are familiar with its administration in your environment, including installing the SiteMinder SDK and configuring the SiteMinder Policy Server.

The focus of this document is to provide details on configuring policy settings for the SiteMinder Policy Server and Web Agent, setting the required security properties on the Integrity Lifecycle Manager server, and enabling secure communications (SSL).

Limitations for Triggers, Integrations, and API

Integrity Lifecycle Manager event triggers function as an internal server-to-server connection. When using the SiteMinder domain, the Integrity Lifecycle Manager server should be configured to use an IP-based policy scheme list through the MKS Domain or through the SiteMinder Direct domain.

Integration types that are supported for SiteMinder authentication only include client-side integrations that use the common session (for example, Microsoft Visual Studio and Eclipse Platform integrations). Other integration types may be used; however, they must be configured to use a different type of authentication.

2

Configuring the SiteMinder Policy Server

Configuration for the Integrity Lifecycle Manager server requires the creation of a Custom Agent. This section outlines the general steps for creating the Custom Agent and configuring the SiteMinder Policy Server to work with the Integrity Lifecycle Manager Agent.

Note

This section is intended for administrative users who are familiar with configuring and managing SiteMinder. The instructions provide general configuration steps for the Integrity Lifecycle Manager integration only and assume that the SiteMinder administrative user already exists. For complete details on configuring SiteMinder, consult the SiteMinder product documentation.

To configure the SiteMinder Policy Server for Integrity Lifecycle Manager

1. Open the SiteMinder Administration (policy server) interface.
2. Configure a new agent type.
   a. Select **Create Agent Type** and assign a name for the new agent type (for example, `IntegrityServer`). This name is used later in the configuration.
   b. Create an action (for example, an action named `authenticate`).
3. Create an agent.
   a. Select **Create Agent** and enter the name of the agent. This name is used later in the configuration when creating the new realm.
   b. Select the option for **Support 4.x agents**.
   c. Set `IntegrityServer` as the agent type.
   d. Set the IP/Host Name for the Integrity Lifecycle Manager server.
   e. Set a password value for the agent's shared secret.
4. Create a new realm under the same SiteMinder domain where the SiteMinder Web Agent resides.
   a. Create a new realm under the domain where the SiteMinder Web Agent resides.
   b. Assign a name for the realm.
   c. Set the agent to the name you specified when you created the agent (in Step 3a).
   d. Set the resource filter to: /
   e. Depending on the authentication to be used, set the authentication scheme to **Basic** or **WebForm** The available selections also depend on the schemes that are configured for your SiteMinder implementation.
   f. Ensure the default resource protection is **Protected**
5. Create a rule under the realm.
   a. Create a new rule under the realm you created in Step 4.
   b. Set the resource to: *
   c. For the action, select the **IntegrityServer actions** option, and then from the list, choose **authenticate**.
   d. Ensure the options for **Allow Access** and **Enabled** are both selected.
6. Create a response.

a. Create a response under the domain where the SiteMinder Web Agent resides.

b. Assign a name for the response.

c. Choose **SiteMinder** as the agent type, and then select **IntegrityServer** from the list.

d. Create an attribute named `authenticate` with the value `success`.

Note

When setting the response attribute, ensure **Attribute Kind** is set to **Static**.

7. Create a policy.

   a. Create a policy under the domain where the SiteMinder Web Agent resides.

   b. Ensure the option for **Enabled** is selected.

   c. Configure users as required.

   d. Add the rule you created in Step 5, and set the response to the response you created in Step 6.

8. Save your changes.

3

Installing the Integrity Lifecycle Manager Server

After configuring the required settings for the SiteMinder Policy Server, the next step is to install the Integrity Lifecycle Manager server. For details on completing the installation, refer to the *Integrity Lifecycle Manager Installation and Upgrading Guide* on the product DVD.

Upgrading

This guide assumes an upgrading path from Integrity 10.x, directly to this release. There are no unique steps required for upgrading to this release.

For general information on upgrading to this release, refer to the *Integrity Lifecycle Manager Installation and Upgrading Guide* available from the PTC Technical Support (http://www.ptc.com/support/integrity.htm).

4

Configuring Integrity Lifecycle Manager Security Properties for SiteMinder

All required security configuration properties are loaded from the `security.properties` file on the Integrity Lifecycle Manager server. This section describes procedures pertaining to authenticating and configuring security properties for the Integrity Lifecycle Manager SiteMinder Server.

Basic and Form Authentication

The Integrity Lifecycle Manager server provides the following methods for authenticating Integrity Lifecycle Manager client systems with SiteMinder:

Authentication Method	Description
Basic	User name and password credentials are submitted through a dialog box. For the required steps, see To configure Basic authentication on page 20
Form	User name and password credentials are submitted through a Web form. If Form authentication is used, then you must reconfigure certain security settings any time you change the form. For the required steps, see To configure Form authentication on page 20

To specify the authentication method Integrity Lifecycle Manager will use, you configure the following property in the `security.properties` file:

```
mks.security.siteminder.client.authentication.type=basic|form
```

Security Domains

In addition to the authentication method, you must configure the required server and client properties to select a security scheme (for example, `siteminder_private` or `siteminderdirect_private`).

⚠ **Caution**

Private security schemes are recommended for production environments. Clear security schemes should only be used for testing purposes.

The Integrity Lifecycle Manager server provides the following security domains for use with SiteMinder:

Security Domains	Description
SiteMinder	Each Integrity Lifecycle Manager client must first obtain a session token from a Web Agent, which is then validated by the Integrity Lifecycle Manager server through SiteMinder.
SiteMinder Dorect	Integrity Lifecycle Manager client systems can connect directly to the Integrity Lifecycle Manager server without first connecting to the Web Agent. Users connect directly to the Integrity Lifecycle Manager server with a user name and password.

Both domains allow either private or clear authentication schemes; however, only private authentication is recommended for production environments. For the required configuration steps, see To connect Integrity Lifecycle Manager Clients through a Web Agent on page 25 or To connect Integrity Lifecycle Manager Client systems directly to the Integrity Lifecycle Manager Server on page 26

Security Policies

When SiteMinder is one of the domains in use, you should not configure multiple default schemes. For example, in the `security.properties` file, you can have:

- A full client IP address for clients connecting from a specific IP address (for example, `mks.security.policy.scheme.1.2.3.4=mksdomain_clear`).
- A partial client IP address for clients connecting from an IP address range, based on removing the final component of the IP address (for example, `mks.security.policy.scheme.1.2.3=ldap_clear`).
- The default scheme list for clients outside the IP address ranges (for example, `mks.security.policy.scheme.default=siteminder_clear`).

⚠ **Caution**

You should not configure `security.properties` with a default scheme list such as `mks.security.policy.scheme.default=siteminder_clear,mksdomain_clear`. This results in multiple attempts to connect through SiteMinder (3 attempts), and then through the MKS Domain (3 attempts). In addition, the displayed login dialog does not advise the user on the specific domain they are connecting to, increasing the chance of login errors.

To configure Basic authentication

1. In a text editor, open the following Integrity Lifecycle Manager server configuration file: `*Integrity Server installdir*/config/properties/security.properties`

   where *`Integrity Server installdir`* is the path to the directory where you installed the Integrity Lifecycle Manager server. By default, the installation path is `Integrity/IntegrityServer10`.
2. Add `mks.security.siteminder.client.authentication.type=basic`
3. Save your changes.

Note

If the value for `mks.security.siteminder.client.authentication.type` is unset or incorrectly set, each connecting Integrity Lifecycle Manager client will use Basic authentication by default.

To configure Form authentication

1. In a text editor, open the following Integrity Lifecycle Manager server configuration file:

   `*Integrity Server installdir*/config/properties/security.properties` where *`Integrity Server Installdir`* is the path to the directory where you installed the Integrity Lifecycle Manager server. By default, the installation path is `Integrity/ILMServer11`.

2. Modify the `security.properties` file as follows:

   - Add `mks.security.siteminder.client.authentication.type=form`
   - Add `mks.security.siteminder.client.form.post.url` = *the absolute address to the URL where the form is directed*

     Note

     If the SiteMinder Web Agent has a form action to an absolute URL, you must set a value for `mks.security.siteminder.client.form.post.url`. The property value can be unset if the form action has a relative URL.

   - Add `mks.security.siteminder.client.form.field.username` = *userID*
   - Add

     `mks.security.siteminder.client.form.field.password` = *password*

3. Save your changes.

Note

If you later change the form that is in use, you must reconfigure these properties and restart the Integrity Lifecycle Manager server.

If the value for `mks.security.siteminder.client.authentication.type` is unset or incorrectly set, each connecting Integrity Lifecycle Manager client will use Basic authentication by default.

To configure general Integrity Lifecycle Manager security properties for SiteMinder

1. In a text editor, open the following Integrity Lifecycle Manager server configuration file:

   `Integrity Server installdir/config/properties/security.properties`

   where `Integrity Server installdir` is the path to the directory where you installed the Integrity Lifecycle Manager server. By default, the installation path is `Integrity/ILMServer11`.

2. Add the following properties:

   - For the Policy Server, set the IP (or DNS) address:

     `mks.security.siteminder.ps.ip=`*SiteMinder Policy Server IP or DNS*

   - For resources protected by the SiteMinder Web Agent, set:

     `mks.security.siteminder.ps.resource.name=`*resource name* (as previously configured in Step 5 under “create a rule under the realm.” in this section: To configure the SiteMinder Policy Server for Integrity Lifecycle Manager on page 12)

     `mks.security.siteminder.ps.resource.action=`*action name* (as previously configured in Step 6 under “Create a response.” in this section: To configure the SiteMinder Policy Server for Integrity Lifecycle Manager on page 12)

     `mks.security.siteminder.client.request.url=`*fully qualified, absolute URL of the resource protected by the SiteMinder Web Agent*

   - For the Custom Agent configured on the Policy Server, set the Agent name to:

     `mks.security.siteminder.ps.agentname=integrityserver` (the agent name for the Custom Agent configured on the Policy Server.)

   - For the Custom Agent configured on the Policy Server, set the Shared Secret:

     `mks.security.siteminder.ps.agentsecret=`*shared secret*

3. Set the default security policy scheme, for example, `siteminder_private`, `siteminderdirect_private`, `MKSDomain`, or other supported schemes. For more information see “Security Policies”
4. If you have configured the SiteMinder Policy Server with the standard defaults for ports, connections, and timeout settings, then no additional modifications are required. If you have not configured the Policy Server with the default settings, then you must also add the following properties in the `security.properties` file:

SiteMinder Direct - Clear Connection Properties (For Testing Only)	Value
`mks.security.siteminder.ps.acport=`	44441
`mks.security.siteminder.ps.auport=`	44442
`mks.security.siteminder.ps.azport=`	44443
`mks.security.siteminder.ps.conmax=`	3
`mks.security.siteminder.ps.conmin=`	1
`mks.security.siteminder.ps.constep=`	1
`mks.security.siteminder.ps.timeout=`	75

5. As required, configure the settings for Integrity Lifecycle Manager client connections through the Web Agent or for direct connections to the Integrity Lifecycle Manager server. For detailed steps, see:
   - To connect Integrity Lifecycle Manager Clients through a Web Agent on page 25
   - To connect Integrity Lifecycle Manager Clients directly to the Integrity Lifecycle Manager Server on page 26
6. To allow the Integrity Lifecycle Manager client to obtain a SiteMinder session token, add the following Integrity Lifecycle Manager server properties for resources protected by SiteMinder:

   `mks.security.siteminder.client.request.url=`*URL of SiteMinder-protected resource*

   `mks.security.siteminder.client.form.field.username=`

```
mks.security.siteminder.client.form.field.password=
```

Caution

The URL of the SiteMinder-protected resource must be enabled for SSL. To enable SSL, you must have a valid, signed server certificate to allow secure communications. For SSL connections, `mks.security.siteminder.client.request.url` must have a URL defined with `https` in the address. For example, `https://apacheserver.com/ siteminder/protected.html.`

If `mks.security.siteminder.client.request.url` is unset then the Integrity Lifecycle Manager client connects directly to the Integrity Lifecycle Manager server . This configuration also works in a reverse proxy setup (where the proxy resides on the Integrity Lifecycle Manager server).

All properties required for the Integrity Lifecycle Manager client can be configured in the server `security.properties` file.

7. Install CA SiteMinder SDK r6.0 SP6 as appropriate for your operating system.

Note

SDK r6.0 SP6 is required whether you are working with SiteMinder 6.0 or 12.5. The SiteMinder SDK is generally installed on the same machine that hosts the Integrity Lifecycle Manager server.

8. From `netegrity/sdk/java` (or from `netegrity/sdk/java64`), copy the `smjavaagentapi.jar` file to the following folder on the Integrity Lifecycle Manager server:

   `Integrity Server installdir``/server/mks/lib/`
9. From `netegrity/sdk/bin` (or from `netegrity/sdk/bin64`), do one of the following (depending on your operating system):
   - On Windows, copy the `DLL` files to *`Integrity Server installdir`*`/server/mks/bin`.
   - On UNIX, copy all `SO` files to *`Integrity Server installdir`*`/server/mks/bin`.
10. On Windows, ensure *`Integrity Server installdir`*`/server/mks/bin` is in the system `PATH`.

On UNIX, ensure *Integrity Server installdir*/server/mks/bin is in the LD_LIBRARY_PATH.

11. For all platforms, edit the isutil.lax file in the *Integrity Server installdir*/bin directory and append:

    ```
    ;..\\server\\mks\\lib\\smjavaagentapi.jar
    ```

    to the line starting with:

    ```
    lax.class.path=
    ```

Note

The lax.class.path property provides the Java classpath required to run the application. The list can use colon separators (on Unix) or semi-colons (on Windows).

12. If required, configure the necessary settings to support Integrity Lifecycle Manager proxies in your SiteMinder environment. For more information, see Configuring FSA for SiteMinder on page 29
13. To apply the changes, start the Integrity Lifecycle Manager server. No special configuration is required for each individual Integrity Lifecycle Manager client.

To connect Integrity Lifecycle Manager client systems through a Web Agent

For Integrity Lifecycle Manager client connections through a Web Agent (siteminder), add the following properties to the security.properties file:

SiteMinder - Private Connection Properties	Value
mks.security.scheme.siteminder_private.description=	Authenticate using password over secure connection
mks.security.scheme.siteminder_private.connectionProvider=	mks.frame.client.SiteMinder PrivateConnectionProvider
mks.security.scheme.siteminder_private.authentication=	siteminder
mks.security.scheme.siteminder_private.realm=	mksdomain

Note

You can configure users and groups from any supported domain. For example, the MKS Domain or LDAP.

SiteMinder - Clear Connection Properties (For Testing Only)	Value
`mks.security.scheme.siteminder_clear.description=`	Authenticate over clear connection
`mks.security.scheme.siteminder_clear.connectionProvider=`	`mks.frame.client.SiteMinder ClearConnectionProvider`
`mks.security.scheme.siteminder_clear.authentication=`	`siteminder`
`mks.security.scheme.siteminder_clear.realm=`	`mksdomain`

Note

The MKS Domain is referenced by default; however, all supported domains for Integrity Lifecycle Manager are also valid.

To connect Integrity Lifecycle Manager Client systems directly to the Integrity Lifecycle Manager Server

For direct Integrity Lifecycle Manager client connections to the Integrity Lifecycle Manager server (`siteminderdirect`) using user names and passwords, add the following properties to the `security.properties` file:

SiteMinder Direct - Private Connection Properties	Value
`mks.security.scheme.siteminderdirect_private.authentication=`	`siteminderdirect`
`mks.security.scheme.siteminderdirect_private.description=`	SiteMinder authentication over private connection

SiteMinder Direct - Private Connection Properties	Value
mks.security.scheme.siteminderdirect_private.connectionProvider=	mks.frame.client.PrivateConnectionProvider
mks.security.scheme.siteminderdirect_private.realm=	mksdomain

SiteMinder Direct - Clear Connection Properties (For Testing Only)	Value
mks.security.scheme.siteminderdirect_clear.authentication=	siteminderdirect
mks.security.scheme.siteminderdirect_clear.connectionProvider=	mks.frame.client.ClearConnectionProvider
mks.security.scheme.siteminderdirect_clear.description=	SiteMinder authentication over clear connection
mks.security.scheme.siteminderdirect_clear.realm=	mksdomain

5

Configuring FSA for SiteMinder

Federated Server architecture (FSA) is an implementation of the Integrity Lifecycle Manager server structured to serve client requests through a proxy. The proxy provides access to project members residing on the Integrity Lifecycle Manager server by retrieving information from its local cache or, if changes are detected, directly from the server.

In the FSA development environment, developers who use configuration management have full visibility into their projects. Users communicate with a proxy, that in turn, communicates with the host server. The proxy stores necessary project information on a disk and/or in a cache, and uses TCP/IP to communicate with the host server for updates.

When a file is checked out, it is checked out from the master repository, though source data may be provided by the proxy. The status of the checked out file is visible to all users regardless of location. When that file is checked in, it is checked in to the master repository so that the most current information is always available.

For FSA to function, certain properties must be configured on the host Integrity Lifecycle Manager server, as well as on the proxy machine. The proxy must also have access to the SiteMinder policy server.

FSA configurations are supported for both the `siteminder` and `siteminderdirect` security schemes.

This section provides configuration information for setting up FSA with SiteMinder.

Note

For complete details on configuring and operating with FSA, see the *Integrity Lifecycle Manager Installation and Upgrading Guide*.

FSA in a SiteMinder Environment

In a SiteMinder environment, one option is to have the proxy authenticating to the Integrity Lifecycle Manager server through the MKS Domain, and individual users authenticating through SiteMinder. To set up this particular SiteMinder configuration, you modify the following properties in the `security.properties` files on the Integrity Lifecycle Manager server:

- `mks.security.policy.scheme.default=siteminder_private`

  or

  `mks.security.policy.scheme.default=siteminderdirect_private`
- `mks.security.policy.scheme.`*`IP address of the proxy machine`*`=mksdomain_private`
- `mks.trustedProxies=`*`IP address of the proxy machine`*

  where
- `mks.security.policy.scheme.`*`IP address of the proxy machine`* allows for a specific security scheme based on an IP address.
- The value of `mks.trustedProxies` notifies the Integrity Lifecycle Manager server that users connecting through the identified proxy are allowed to authenticate using schemes identified by the IP of the Integrity Lifecycle Manager client. For example, if `mks.trustedProxies` is set, users can be authenticated via `siteminder_private`. If `mks.trustedProxies` is unset, users are authenticated through `mksdomain_private`, according to the IP of the proxy.

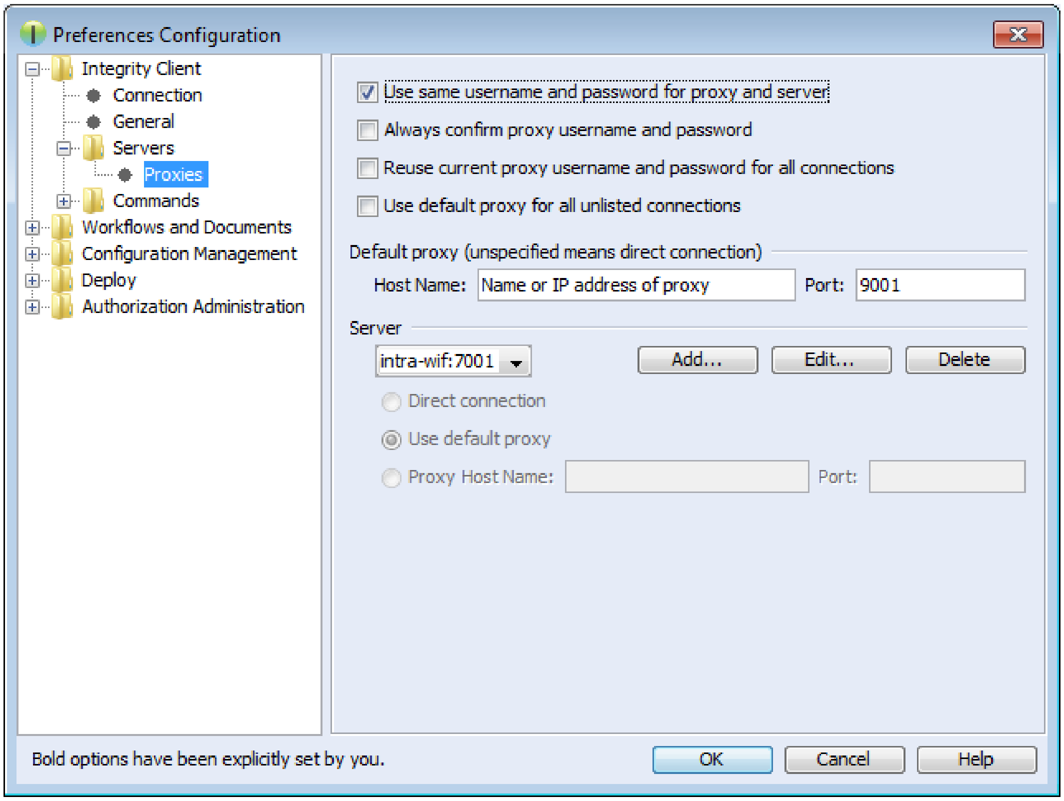
Note

- The `mks.trustedProxies` property is supported only for use with `siteminder` realms.
- For multiple proxies, the value of `mks.trustedProxies` uses a comma-separated list of IP addresses.

Integrity Lifecycle Manager Client Connection Preferences for FSA

When the Integrity Lifecycle Manager client is connecting to the Integrity Lifecycle Manager server through a proxy, and that client does not have access to the server's Web Agent because of network limitations, the client must be able to authenticate to a separate Web Agent that is in the same SiteMinder domain as the Integrity Lifecycle Manager server’s Web Agent.

In this situation, the Integrity Lifecycle Manager client user must configure their client preferences to use the same user name and password for both the proxy and Integrity Lifecycle Manager server, as illustrated in the following **Preferences Configuration** dialog box:



☑ Use same username and password for proxy and server

☐ Always confirm proxy username and password

☐ Reuse current proxy username and password for all connections

☐ Use default proxy for all unlisted connections

Default proxy (unspecified means direct connection)

Host Name: Name or IP address of proxy Port: 9001

Server

intra-wif:7001

○ Direct connection

● Use default proxy

○ Proxy Host Name: Port:

Bold options have been explicitly set by you.

If the Integrity Lifecycle Manager client has access to both Web agents (that is, the Web Agent for the proxy and the Web Agent for the Integrity Lifecycle Manager server) then users do not need to set the option for **Use same username and password for proxy and server**.

6

Enabling SSL Communications

The Secure Sockets Layer (SSL) protocol of the Integrity Lifecycle Manager server enables encrypted, authenticated communication. When users connect through SSL, the connection ensures privacy, authentication, and message Integrity Lifecycle Manager. SSL configurations are supported for both the `siteminder` and `siteminderdirect` security schemes.

In an SSL connection, the server must have a security certificate. Each side then encrypts the data it sends, ensuring the information can only be read by the intended recipient.

The Integrity Lifecycle Manager server complies with US Encryption Export Control Regulations. By default, the encryption strength for the SSL is 128-bit encryption.

When SSL is enabled for the server, the Web browser client confirms that the:

- certificate has been signed by a recognized Certificate Authority (CA)
- current date falls within the certificate's valid date range

  When generating a new certificate, the following are possible scenarios for using server certificates:

- Certificate is signed by a recognized CA.
- Certificate is signed by CA that may not be well known.
- You are using the new certificate as self-signed.

Note

You must have the signed certificate available when enabling SSL connections or the Integrity Lifecycle Manager server cannot start.

For a configuration with SiteMinder, the administrator must obtain certificates for the machines running the Integrity Lifecycle Manager server and the SiteMinder Web Agent (since the SiteMinder Web Agent must also run on a machine that is configured to receive SSL connections).

This configuration enables the required SSL connections between the Integrity Lifecycle Manager client and the Integrity Lifecycle Manager server, and between the Integrity Lifecycle Manager client and the SiteMinder Web Agent.

Note

The secure server running the SiteMinder Web Agent should have a certificate from a recognized CA; however, if the certificate is from a CA that is not well known, or if the certificate is self-signed, then each Integrity Lifecycle Manager client must have that certificate imported to the list of trusted certificates in the client's JVM. For more information, see To import a certificate to the Integrity Lifecycle Manager Client list of trusted certificates on page 35

To enable SSL connections in a SiteMinder configuration

1. For the Integrity Lifecycle Manager server, configure the appropriate property keys in:

   `Integrity Server installdir``/config/properties/is.properties`

   where *`Integrity Server installdir`* is the Integrity Lifecycle Manager server installation directory.

2. To enable SSL, set the following property key:

   `mksis.secure.port=`*`SSL port number`*

   Note

   A value of *`0`* disables the SSL connection.

3. Set a password for the following property key:

   `mksis.privatekey.password=`*`keystore password`*

   where *`keystone password`* is the password used during certificate creation.

   Tip

   For details on creating a signed certificate, see the Integrity Lifecycle Manager Online Help.

4. Set the certificate for the SiteMinder Web Agent. For more information, refer to the SiteMinder product documentation.

To import a certificate to the Integrity Lifecycle Manager client list of trusted certificates

1. Navigate to the *`Integrity Client installdir`*`/jre/bin` directory.
2. From that location, run the following command:

```
keytool –import –alias unique name –keystore ../lib/
security/cacerts –file Path to the certificate file used
by SiteMinder Web Agent
```

7

Diagnostics

This section provides a summary of available logging functionality, as well as information on diagnosing the Integrity Lifecycle Manager server- SiteMinder configuration.

Logging

CLI Command:

```
im logging --category=SITEMINDER --on --target=
server|client
```

This release includes the following additional logging output for SiteMinder implementations:

Log File	Logging Information
Integrity Server installdir/log/server.log	New SITEMINDER logging category. Provides general debugging information at the MEDIUM(5) logging level. Detailed information, such as token content, is reported at the LOW(10) logging level. The SITEMINDER logging category is also available in the IntegrityClient.log file under: *Integrity Client installdir*/bin
Integrity Server installdir/log/smUtil.log	Provides connection validation information

Checking Your Configuration

To confirm that the Integrity Lifecycle Manager server configuration is correct, you can either connect using an Integrity Lifecycle Manager client or you can run the `isutil -c smUtil` command:

```
isutil -c smUtil policyConnect
```

The command reads the user name and password from the following options:

```
USER_NAME=
USER_PWD=
```

For example:

```
isutil -c smUtil policyConnect USER_NAME=user USER_PWD=
userPassword
```

Note

If the password is not provided, you are prompted to provide it.

Settings in the `security.properties` file can be further modified using the `isutil` command.

For example:

```
isutil -c smUtil policyConnect USER_NAME=user USER_PWD=
userPassword
```

```
mks.security.siteminder.ps.agentname=newIntegrityServer
```