Web Policy Agents Guide

Sun™ ONE Identity Server Policy Agents

Version 2.1

816-6772-10
April 2005

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.

Contents

About This Guide

Sun™ ONE Identity Server Policy Agents, version 2.1 comprise Web Policy Agents and J2EE Policy Agents. This guide offers an introduction to Web Policy Agents and describes how to install and configure these agents on web servers and proxy servers.

This preface contains the following sections:

What You Are Expected to Know

This book is considered to be an auxiliary manual in the documentation series provided with Sun ONE Identity Server. It's essential that you understand directory technologies and have some experience with Java and XML programming languages. You will get the most out of this guide if you are familiar with directory servers and Lightweight Directory Access Protocol (LDAP). Particularly, you should be familiar with Sun ONE Directory Server and the documentation provided with that product.

This guide is intended for use by IT professionals who manage access to their network through Sun ONE servers and services. The functionality contained in Sun ONE Identity Server allows you to manage user data and enforce access policies throughout your enterprise.

As you try to understand the concepts described in this guide, you should reference the *Sun ONE Identity Server Installation and Configuration Guide* and the *Sun ONE Identity Server Programmer's Guide*.

Identity Server's Documentation Set

The Sun ONE Identity Server documentation set contains the following titles:

- *Product Brief* provides an overview of the Sun ONE Identity Server and its features and functions.
- *Installation Guide* provides details on how to install and deploy Sun ONE Identity Server on Solaris™, Linux and Windows® 2000 systems.
- *Administration Guide* describes how to use the Sun ONE Identity Server console as well as manage user and service data via the command line.
- *Programmer's Guide* documents how to customize an Identity Server system specific to your organization. It also includes instructions on how to augment the application with new services using the public APIs.
- *Getting Started Guide* documents how to use various features of Sun ONE Identity Server to set up a simple organization with identities, policies and roles.
- *J2EE Policy Agents Guide* documents how to install and configure Sun ONE Identity Server policy agents for application servers on a remote server. It also includes troubleshooting and information specific to each agent.
- *Web Policy Agents Guide* (this guide) documents how to install and configure Sun ONE Identity Server policy agents for web servers on a remote server. It also includes troubleshooting and information specific to each agent.
- The *Release Notes* file gathers an assortment of last-minute information, including a description of what is new in this release, known problems and limitations, installation notes, and how to report problems.

NOTE Be sure to check the Sun ONE Identity Server documentation web site for updates to the release notes and for revisions to the guides. They are available at `http://docs.sun.com/db/prod/s1idsrv#hic`. Updated documents will be marked with a revision date.

Documentation Conventions Used in This Manual

In the Sun ONE Identity Server 6.0 SP1 documentation set (such as this guide) there are certain typographic and terminology conventions used to simplify discussion and to help you better understand the material. These conventions are described below.

Typographic Conventions

This book uses the following typographic conventions:

- *Italic type* is used within text for book titles, new terminology, emphasis, and words used in the literal sense.
- `Monospace font` is used for sample code and code listings, API and language elements (such as function names and class names), filenames, pathnames, directory names, HTML tags, and any text that must be typed on the screen.
- *Italic serif font* is used within code and code fragments to indicate variable placeholders. For example, the following command uses *filename* as a variable placeholder for an argument to the gunzip command:

  `gunzip -d` *filename*`.tar.gz`

Terminology

Below is a list of the general terms that are used in the Sun ONE Identity Server Policy Agent documentation:

- *Agent_Install_Dir* is a variable placeholder for the directory where you have installed the Sun ONE Identity Server Policy Agent.
- *S1IS_Install_Dir* is a variable placeholder for the home directory where you have installed Sun ONE Identity Server 6.0.

Related Third-Party Web Site References

Sun is not responsible for the availability of third-party Web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Related Information

In addition to the documentation provided with Sun ONE Identity Server, there are several other sets of documentation that might be helpful. This section lists these and additional sources of information.

iPlanet Directory Server Documentation

iPlanet Directory Server 5.1 documentation can be found at
http://docs.sun.com/db/coll/S1_ipDirectoryServer_51

iPlanet/Sun ONE Web Server Documentation

iPlanet/Sun ONE Web Server documentation can be found at
http://docs.sun.com/db/coll/S1_ipwebsrvree60_en

iPlanet Proxy Server Documentation

iPlanet Proxy Server documentation can be found at
http://docs.sun.com/db/coll/S1_ipwebproxysrvr36

Other iPlanet Product Documentation

Documentation for all other Sun ONE servers and technologies can be found at
http://docs.sun.com/prod/ds/sunone

Download Center

Links to download any of Sun's Sun ONE/iPlanet software are at
http://wwws.sun.com/software/download/

Sun ONE Technical Support

Technical Support can be contacted through
http://www.sun.com/service/support/software/iplanet/index.html

Professional Services Information

Professional Service can be contacted through
http://www.sun.com/service/sunps/iplanet/

Sun Enterprise Services for Solaris Patches And Support

Solaris patches and support can be obtained through
http://www.sun.com/service/

Developer Information

Information on Sun ONE Identity Server, LDAP, the Sun ONE Directory Server, and associated technologies can also be found at
http://developer.iplanet.com/tech/directory/

Chapter 1

Read This First

Sun™ ONE Identity Server Policy Agents, version 2.1 comprises Web Policy Agents and J2EE Policy Agents. This chapter provides a brief overview of Web Policy Agents as well as some concepts you will need to understand before proceeding with the installation of these agents. The information in this chapter is common to all the supported operating systems.

Topics in this chapter include:

- Web Policy Agents
- Supported Servers
- Before You Begin Installation

In related documentation, you might see Sun ONE Identity Server referred to as Sun Java™ System Identity Server and Sun Java™ System Access Manager. These three names refer to the same product, but different versions.

Web Policy Agents

Web Policy Agents protect content on web servers and web proxy servers from unauthorized intrusions. They control access to services and web resources based on the policies configured by an administrator.

Uses for Web Policy Agents

Policy agents are installed on web servers for a variety of reasons. Here are three examples:

- An agent on a human resources server prevents non-human resources personnel from viewing confidential salary information and other sensitive data.
- An agent on an Operations web server allows only network administrators to view network status reports or to modify network administration records.
- An agent on an Engineering web server allows authorized personnel from many internal segments of a company to publish and share research and development information. At the same time, the agent restricts external partners from gaining access to the proprietary information.

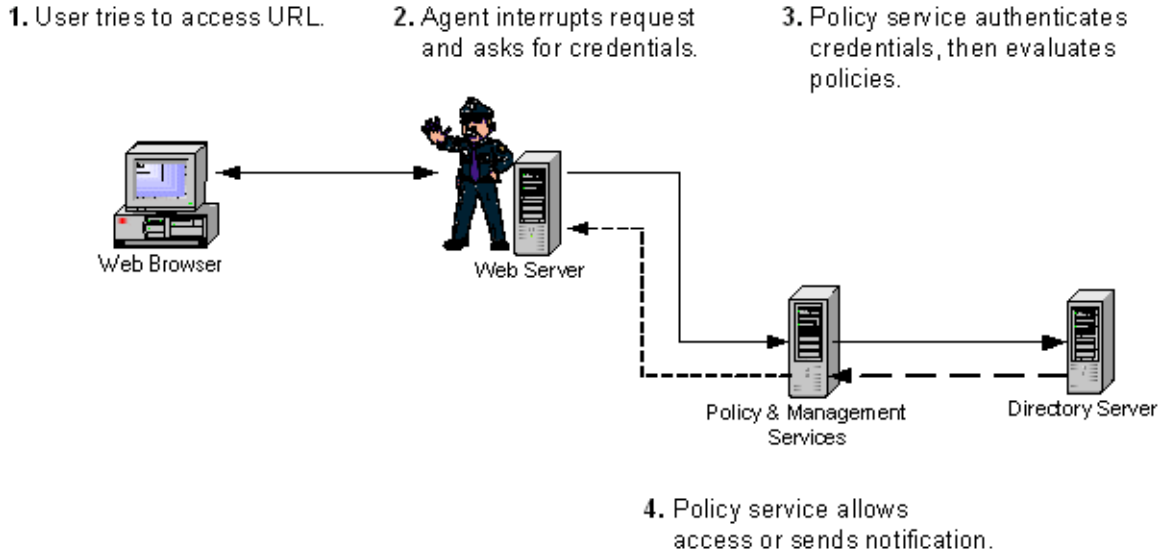
In each of these situations, a system administrator must set up policies that allow or deny users access to content on a web server. For information on setting policies and for assigning roles and policies to users, see the *Sun ONE Identity Server Administration Guide*.

How an Agent Interacts With Sun ONE Identity Server

Figure 1-1 illustrates how a policy agent installed on a remote web server interacts with Sun ONE Identity Server. When a user points a browser to a particular URL on a protected web server, the following interactions take place:

1. The agent intercepts the request and validates the existing authentication credentials. If the existing authentication level is insufficient, the appropriate Sun ONE Identity Server authentication service will present a login page. The login page prompts the user for credentials such as username and password.
2. The authentication service verifies that the user credentials are valid. For example, the default LDAP authentication service verifies that the username and password are stored in Sun ONE Directory Server. You might use other authentication modules such as RADIUS and Certificate modules. In such cases, credentials are not verified by Directory Server but are verified by the appropriate authentication module.
3. If the user's credentials are properly authenticated, the policy agent examines all the roles assigned to the user.
4. Based on the aggregate of all policies assigned to the user, the individual is either allowed or denied access to the URL.

Figure 1-1 An Agent's Interaction With Sun ONE Identity Server



Supported Servers

Web Policy Agents, version 2.1 support the following web and proxy servers. This version of the web policy agents supports Sun ONE Identity Server, versions 6.0 SP1, 6.1 and 6.2. Note that the agent supported on Solaris 8 platform is generally also supported on Solaris 9 platform and vice versa.

Table 1-1 Supported Servers and Platforms

Agent for	Supported Platform
Sun ONE Web Server 6.0 SP*x*	Solaris 8
	Solaris 9
	Microsoft Windows 2000
	HP-UX 11.11
Sun ONE Web Proxy Server 3.6 (in reverse proxy mode)	Solaris 8
Apache 1.3.27	Solaris 8
	Solaris 9
	Red Hat Linux 7.2

Agent for	Supported Platform
IBM Lotus Domino 6.0.1	Solaris 8
	Microsoft Windows 2000
IBM Lotus Domino 5.0.11	Solaris 8
	Microsoft Windows 2000
IBM HTTP Server 1.3.19	Solaris 8
Microsoft IIS 5.0	Microsoft Windows 2000
Apache 2.0.47	Red Hat Linux 9.0
IBM HTTP Server 1.3.26	Solaris 8
	Solaris 9
Sun ONE Web Server 4.1	Solaris 8
Oracle9iAS Apache 1.3.29	Solaris 8
SAP Internet Transaction Server 2.0	Microsoft Windows 2000 Advanced Server
Apache 2.0.48	Solaris 8
	Solaris 9
	Red Hat Advanced Server 2.1
Microsoft IIS 6.0	Microsoft Windows Server 2003 EE
Sun Java System Web Server 6.1	Solaris 8
	Solaris 9 x86
	Microsoft Windows 2000
	Microsoft Windows Server 2003 EE
Oracle 10g Apache Server	Solaris 9
IBM Lotus Domino 6.5	Solaris 9
	Microsoft Windows 2000
	Microsoft Windows Server 2003 EE
IBM Lotus Domino 6.5.1	Red Hat Advanced Server 2.1
Apache 2.0.50	Microsoft Windows Server 2003 EE
Apache 1.3.29	SuSE Linux 8.2
Apache 2.0.52	Debian Linux 3.0
	SuSE Linux 8.2

Before You Begin Installation

Read the following sections carefully before you start the installation program:

Java Runtime Environment 1.3.1 or Higher

You must have Java Runtime Environment (JRE) 1.3.1 or higher installed or available on a shared file system in order to run the graphical user interface (GUI) of the agent installation program. Currently, JRE 1.3.1 or any version higher is certified for use with the agent installation program. See “Installing From the Command Line” for more information.

If you are running the installation program on the Windows operating system, the installation program will install JRE 1.3.1 or higher if it does not detect JRE on the system.

Remote Web Servers

You can use the installation program to install a policy agent on the web server where Sun ONE Identity Server is installed. In Sun ONE documentation, this server is referred to as the *web server that runs the Sun ONE Identity Server*. You can also use the installation program to install additional policy agents on remote web servers in your enterprise. A *remote* web server in a Sun ONE Identity Server deployment is any web server other than the one that runs Sun ONE Identity Server. It is "remote" relative to the Sun ONE Identity Server's dedicated web server.

Configuring the Agent for Multiple Web Server Instances on the Same Computer

If you have multiple web server or proxy server instances installed on one computer system, you can configure the agent for each server instance.

For information on how to do this, see "Configuring the Agent for Multiple Web Server Instances."

NOTE Only one instance of Microsoft IIS server can be installed per computer system; you cannot install multiple Microsoft IIS agents on the same computer system.

Providing Failover Protection for Agents

When you install a policy agent, you can specify a *failover* or backup web server for running Sun ONE Identity Server. This is essentially a high availability option. It ensures that if the web server that runs Sun ONE Identity Server service becomes unavailable, the agent can still process access requests through the secondary or the failover web server running Sun ONE Identity Server service.

To set up failover protection for the policy agent, you must first install two different instances of Sun ONE Identity Server on two separate web servers. See the detailed instructions in the *Sun ONE Identity Server Installation and Configuration Guide* to do this and then follow the instructions in the subsequent chapters of this guide to install the appropriate agent. The agent installation program will prompt you for the host name and port number of the failover web server that you have configured to work with Sun ONE Identity Server. The following property in the `AMAgent.properties` file stores the failover server name and port:

com.sun.am.policy.am.loginURL= http://*primary_Identity_Server.example*.com:58080/amserver/UI/Login http://*failover_Identity_Server.example*.com:58080/amserver/UI/Login

The failover server name is configurable after it has been set during the installation. In the properties file, it is the second entry in this property following the primary Sun ONE Identity Server login URL separated by a space.

Updating the Agent Cache

Each agent maintains a cache that stores the policies for every user's session. The cache can be updated by either a cache expiration mechanism or a notification mechanism.

Cache Updates

The agent maintains a cache of all active sessions. Once an entry is added to the cache, it remains valid for a period of time after which the entry is considered expired and later purged.

The property com.sun.am.policy.am.cacheEntryLifeTime in AMAgent.properties determines the number of minutes an entry will remain in the agent cache. Once the interval specified by this property has elapsed, the entry is dropped from the cache. By default, the expiration time is set to three minutes.

Hybrid Cache Updates

In this mode, cache entry expiration still applies. In addition, the agent gets notified by the Sun ONE Identity Server service about session changes. Session changes include events such as session logout or a session timeout. When notified of a session or a policy change, the agent updates the corresponding entry in the cache. Apart from session updates, agents can also receive policy change updates. Policy changes include events such as updating, deleting, and creating policies.

Sun ONE Identity Server web policy agents have the hybrid cache update mode switched on by default. This is triggered by the property com.sun.am.policy.am.notificationEnabled in the AMAgent.properties file, which is set to true. When the property is set to false, the agent updates its cache through cache entry expiration mechanism only.

Restrictions due to firewalls, as well as the type of web server in use, might not allow notifications to work. In such cases, notification is turned off.

The agent sets a timeout period on its cache entries. After its end of life, the cache entry is purged from the agent's cache. The agent does not refetch the cache data. The next attempt to access the same entry from cache fails and the agent makes a round trip to the server and fetches it again to populate the cache. This lazy method of cache updating keeps the agent cache performing optimally and reduces network traffic.

In a normal deployment situation, policy changes on the server are frequent, which requires sites to accept a certain amount of latency for agents to reflect policy changes. Each site decides the amount of latency time that is acceptable for the site's specific needs. When setting the `cacheEntryLifeTime` property, set it the lower of the two:

- The session idle timeout period
- Your site's accepted latency time for policy changes

NOTE The notification support is not available:

- For Apache 1.3.27 agent on all platforms
- For Apache 2.0.47 on Red Hat Linux 9.0
- If the Microsoft IIS 5.0 agent is using HTTPS

Not-Enforced URL List

The *not-enforced URL list* defines the resources that should not have any policies (neither allow nor deny) associated with them.

By default, the policy agent denies access to all resources on the web server that it protects. However, various resources available through a web server (such as a web site or an application) might not need to have any policy enforced. Common examples of such resources include the HTML pages and `.gif` images found in the home pages of web sites. The user should be able to browse such pages without authenticating. These resources need to be on the not-enforced URL list. The property `com.sun.am.policy.agents.notenforcedList` will be used for this purpose. Wild cards can be used to define a pattern of URLs. Space is the separator between the URLs mentioned in the list.

There can be a reverse scenario when all the resources on the web server, except a list of URLs, are open to any user. In that case, the property `com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList` would be used to reverse the meaning of `com.sun.am.policy.agents.notenforcedList`. If it is set to `true` (by default it is set to `false`), then the not-enforced URL list would become the enforced list.

Here are a few examples:

Scenario 1:

```
com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList=
false
```

```
com.sun.am.policy.agents.notenforcedList =
http://mycomputer.example.com:80/welcome.html
http://mycomputer.example.com:80/banner.html
```

In this case, authentication and policies will not be enforced on the two URLs listed in the `notenforcedList`. All other resources will be protected by the agent.

Scenario 2:

```
com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList=
true
```

```
com.sun.am.policy.agents.notenforcedList =
http://mycomputer.example.com:80/welcome.html
http://mycomputer.example.com:80/banner.html
```

In this case, authentication and policies will be enforced by the agent on the two URLs mentioned in the `notenforcedList`. All other resources will be accessible to any user.

Not-Enforced IP Address List

The `com.sun.am.policy.agents.notenforced_client_IP_address_list` property is used to specify a list of IP addresses. No authentication is required for the requests coming from these client IP addresses.

In other words, the agent will not enforce policies for the requests originating from the IP addresses in the Not-Enforced IP Address list.

Enforcing Authentication Only

The property `com.sun.am.policy.agents.do_sso_only` is used to specify if only authentication is enforced for URLs protected by the agent. If this property is set to `true` (by default it is set to `false`), it indicates that the agent enforces authentication only, without enforcing policies. After a user logs onto Identity Server successfully, the agent will not check for policies related to the user and the accessed URLs.

Forwarding LDAP User Attributes via HTTP Headers

The policy agent has the ability to forward LDAP user attribute values via HTTP headers to end-web applications. The LDAP user attribute values come from the server side of Sun ONE Identity Server. The policy agent behaves like a broker to obtain and relay user attribute values to the destination servlets, CGI scripts, or ASP pages. These applications can in turn use the attribute values to personalize page content.

This feature is configurable through two properties in the `AMAgent.properties` file. To turn this feature on and off, use the following property from the `AMAgent.properties` file:

`com.sun.am.policy.am.ldapattribute.mode`

This property can be set to one of the following values:

- `NONE`
- `HEADER`
- `COOKIE`

When set to `NONE`, the agent does not fetch LDAP attributes from the server and ignores the `headerAttributes` property. In the other two cases, agents fetch the attribute.

To configure the attributes that are to be forwarded in the HTTP headers, use the following property:

`com.sun.am.policy.am.headerAttributes`

Below is an example section in the `AMAgent.properties` file, which shows how this feature is used:

```
#
# The policy attributes to be added to the HTTP header.  The
# specification is of the format
# ldap_attribute_name|http_header_name[,...]. ldap_attribute_name
# is the attribute in data store to be fetched and
# http_header_name is the name of the header to which the value
# needs to be assigned.
#
# NOTE: In most cases, in a destination application where a
# "http_header_name" shows up as a request header, it will be
# prefixed by HTTP_, and all lower case letters will become upper
# case, and any - will become _; For example, "common-name" would
# become "HTTP_COMMON_NAME"
#
com.sun.am.policy.am.headerAttributes=cn|common-name,ou|organizational-unit
,o|organization,mail|email,employeenumber|employee-number,c|country
```

By default, some LDAP user attribute names and HTTP header names are set to sample values.

To find the appropriate LDAP user attribute names, check the following XML file on the machine where Sun ONE Identity Server is installed:

S1IS_Install_Dir/SUNWam/config/xml/amUser.xml

The attributes in this file could be either Identity Server user attributes or Identity Server dynamic attributes. For explanation of these two types of user attributes, refer *Sun ONE Identity Server Administration Guide.*

The attribute and HTTP header names that need to be forwarded must be determined by the end-user applications on the web server that the agent is protecting. After all, these applications are the consumers of the forwarded header values (the forwarded information is used for the customization and personalization of web pages).

NOTE The header attributes are available for all web servers except Lotus Domino (all versions).

The Agent Properties File

The `AMAgent.properties` file stores configuration parameters used by the policy agent. From time to time, you may need to make changes to the default parameters in this file. For example, when you want to specify a different failover web server for running Sun ONE Identity Server.

The `AMAgent.properties` file includes information for the following configurations:

- debugging
- policy agent
- FQDN map
- Sun ONE Identity Server services
- service and agent deployment descriptors
- session failover

The `AMAgent.properties` file also contains configuration information on advanced features, such as forwarding LDAP user attributes through `HTTP` headers and POST data preservation. The file has comments before each property; refer to the file for more details. Appendix A of this guide also provides detailed description of each of the properties in this file.

Table 1-2 provides the default location for `AMAgent.properties` on the various supported servers.

Table 1-2 Locating `AMAgent.properties` on Different Platforms

Server	Location
All supported UNIX web servers	`/etc/opt/SUNWam/agents/`*WebServer*`/config/`*_PathInstanceName*`/` where *WebServer* refers to one of the following: •es6 •proxy •apache •ibmhttp •domino •domino6
Sun ONE Web Server 6.0 Microsoft Windows 2000	`\`*Agent_Install_Dir*`\es6\config\`*_PathInstanceName*`\`

Table 1-2 Locating `AMAgent.properties` on Different Platforms *(Continued)*

Server	Location
Lotus Domino 5.0.11 or 6.0.1 Microsoft Windows 2000	*Agent_Install_Dir*\domino\config_*PathInstanceName*\
Microsoft IIS 5.0 Microsoft Windows 2000	*Agent_Install_Dir*\iis\config_*PathInstanceName*\
Microsoft IIS 6.0 Microsoft Windows Server 2003 EE	*Agent_Install_Dir*\iis6\config_Identifier_*IdentifierNumber*
Apache 2.0.50 Microsoft Windows Server 2003 EE	*Agent_Install_Dir*\apache\config\apache_*PortNumber*

Changing the `AMAgent.properties` file can have serious and far-reaching effects. Remember that you can safely change many of the properties in this file by simply reinstalling the agent. However, if you must make manual changes, keep the following in mind:

- Make a backup copy of this file before you make changes.
- Trailing spaces are significant; use them judiciously.
- Use forward slash (/) to separate directories, not backlash (\). This holds true even on Windows systems.
- Spaces in the Windows file names are allowed.

NOTE If you make changes to the `AMAgent.properties` file, you must restart the web server to make your changes take effect.

Setting the Fully Qualified Domain Name

To ensure appropriate user experience, it is necessary that the users access resources protected by the agent using valid URLs. The configuration property `com.sun.am.policy.agents.fqdnDefault` provides the necessary information needed by the agent to identify if the user is using a valid URL to access the protected resource. If the agent determines that the incoming request does not have

a valid hostname in the URL, it redirects the user to the corresponding URL with a valid hostname. The difference between the redirect URL and the URL originally used by the user is only the hostname, which is changed by the agent to a fully qualified domain name (FQDN) as per the value specified in this property.

This is a required configuration property without which the web server may not start up correctly. This property is set during the agent installation and must not be modified unless absolutely necessary to accommodate deployment requirements. An invalid value for this property can result in the web server becoming unusable or the resources becoming inaccessible.

The property `com.sun.am.policy.agents.fqdnMap` provides another way by which the agent can resolve partial or malformed access URLs and take corrective action. The agent gives precedence to the entries defined in this property over the value defined in the `com.sun.am.policy.agents.fqdnDefault` property. If none of the entries in this property matches the hostname specified in the user request, the agent uses the value specified for `com.sun.am.policy.agents.fqdnDefault` property.

The `com.sun.am.policy.agents.fqdnMap` property can be used for creating a mapping for more than one hostname. This may be the case when the web server protected by this agent is accessible by more than one hostname. However, this feature must be used with caution as it can lead to the web server resources becoming inaccessible.

This property can also be used to override the behavior of the agent in cases where necessary. The format for specifying the property `com.sun.am.policy.agents.fqdnMap` is:

```
com.sun.am.policy.agents.fqdnMap =
[invalid_hostname|valid_hostname][,...]
```

where:

`invalid_hostname` is a possible invalid hostname such as partial hostname or an IP address that the user may provide .

`valid_hostname` is the corresponding valid hostname that is fully qualified. For example, the following is a possible value specified for hostname `xyz.domain1.com`:

```
com.sun.am.policy.agents.fqdnMap = xyz|xyz.domain1.com,
xyz.domain1|xyz.domain1.com
```

This value maps `xyz` and `xyz.domain1` to the FQDN `xyz.domain1.com`.

This property can also be used in such a way that the agent uses the name specified in this map instead of the web server's actual name.

Say you want your server to be addressed as *xyz.hostname*.com whereas the actual name of the server is *abc.hostname*.com. The browser only knows *xyz.hostname*.com and you have specified policies using *xyz.hostname*.com at the Identity Server console. In this file, set the mapping as `com.sun.am.policy.agents.fqdnmap` = `valid` | *xyz.hostname*.com

Cookie Reset Feature

This feature enables the policy agent to reset some cookies in the browser session while redirecting to Identity Server for authentication.

This feature is configurable through two properties in the `AMAgent.properties` file.

- Enable Cookie Reset

  `com.sun.am.policy.agents.cookie_reset_enabled=true`

  This property must be set to `true`, if this agent needs to reset cookies in the response while redirecting to Identity Server for authentication. By default, this is set to `false`.

- Cookie List

  This property gives the comma-separated list of cookies that need to be reset in the response while redirecting to Identity Server for authentication. This property is used only if the Cookie Reset feature is enabled.

  Cookie details must be specified in the following format:

  name[=value][;Domain=value]

  For example,

  `com.sun.am.policy.agents.cookie_reset_list=LtpaToken, cookie1=`*value1*`, cookie2=`*value2*`;Domain=`*example*`.com`

Configuring CDSSO

The Cross Domain Single Sign-On (CDSSO) feature is configurable through three properties in the file `AMAgent.properties`. To turn this feature on or off, use the following property in `AMAgent.properties`:

`com.sun.am.policy.agents.cdsso-enabled=true`

By default, this property is set to `false`, and the feature is turned off. To turn on CDSSO, set this property to `true`.

Set the URL where CDC controller is installed by specifying the URL in the following property:

```
com.sun.am.policy.agents.cdcservletURL =
http://nila.eng.example.com:58080/amserver/cdcservlet
```

The third property, `com.sun.am.policy.agents.cookieDomainList` allows you to specify a list of domains in which cookies have to be set in a CDSSO scenario. This property is used only if CDSSO is enabled. If you leave this property blank, then the fully qualified cookie domain for the agent server will be used for setting the cookie domain. In such a case, it is a host cookie and not a domain cookie.

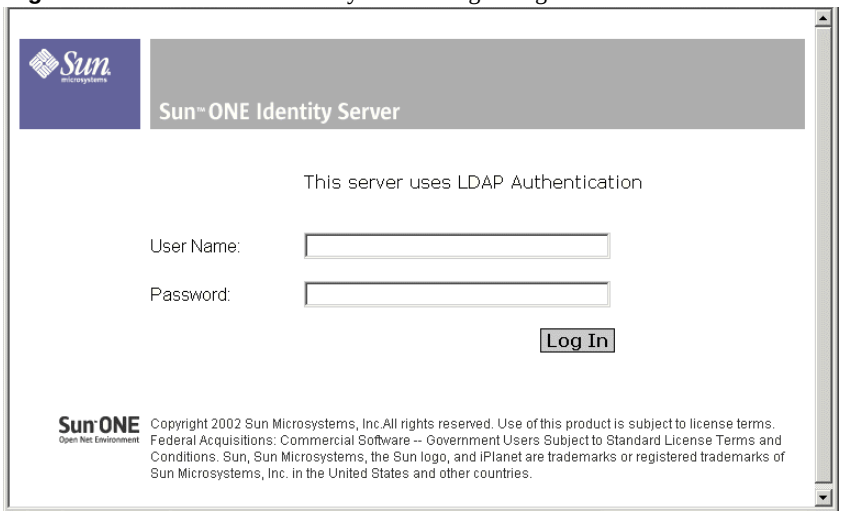
For more information on configuring the CDSSO component, refer *Sun ONE Identity Server Installation Guide.*

Verifying a Successful Installation

After installing a policy agent, it is a good practice to make sure that the agent is installed successfully. There are two things that you can check to verify a successful agent installation.

1. Access some web content on the web server where the agent is installed. If the agent is installed correctly, you should see the Sun ONE Identity Server login page. Figure 1-2 is an example of the Identity Server login page that uses LDAP authentication.

Figure 1-2 Sun ONE Identity Server Login Page



2. Check the file `AMAgent.properties`. Make sure that each property is set properly. For details about the properties in this file, see Appendix A, "AMAgent Properties."

Chapter 2

Policy Agents on Solaris and HP-UX

Sun™ ONE Identity Server Policy Agents work in tandem with Sun™ ONE Identity Server to control user access to web servers in an enterprise. This chapter explains how to install and configure the Web Policy Agents available for the web and proxy servers running on Solaris 8 and 9 and HP-UX 11.11 operating systems.

Topics in this chapter include:

Before You Begin

Be sure that you are familiar with the concepts presented in Chapter 1, “Read This First.” The chapter includes brief but important information on the following topics:

- Supported Servers
- Web Policy Agents
- Java Runtime Environment 1.3.1 or Higher
- Remote Web Servers
- Configuring the Agent for Multiple Web Server Instances on the Same Computer
- Providing Failover Protection for Agents
- Updating the Agent Cache
- Not-Enforced URL List
- Not-Enforced IP Address List
- Enforcing Authentication Only
- Forwarding LDAP User Attributes via HTTP Headers
- The Agent Properties File
- Setting the Fully Qualified Domain Name
- Configuring CDSSO

Installing the Agent

The agent installation program has two interfaces: the Graphical User Interface (GUI) and the Command-Line Interface (CLI). The following sections present instructions to install the agent using both these interfaces.

Installation Using the GUI

Use the following instructions to install the agent using the GUI on the Solaris and HP-UX operating systems.

Installing the Policy Agent for a Web Server

You must have root permissions when you run the agent installation program.

1. Unpack the product binary using the following command:

   ```
   # gunzip -dc binaryname.tar.gz| tar -xvof -
   ```

2. Run the `setup` program. You'll find the program in the directory where you untarred the binaries. At the command line, enter the following:

   ```
   # ./setup
   ```

3. Set your JAVAHOME environment variable to a JDK version 1.3.1_04 or higher. The installation requires that you set up your JAVAHOME variable correctly. However, if you have incorrectly set the JAVAHOME variable, the `setup` script will prompt you for supplying the correct JAVAHOME value:

   ```
   Please enter JAVAHOME path to pick up java:
   ```

4. Type the full path to the directory where JDK is located. The installation program starts with the Welcome page.

5. In the Welcome page, click Next.

6. Read the License Agreement. Click Yes to agree to the license terms.

7. In the Select Installation Directory panel, specify the directory where you would like to install the agent.

   Install Sun ONE Identity Server Policy Agent in this directory: Enter the full path to the directory where you want to install the agent. The default installation directory is `/opt`.

8. Click Next and provide the following information about the web server the agent will protect:

   Host Name: Enter the FQDN of the machine where the web server is installed. For example, *mycomputer*`.`*eng*`.`*example*`.com`

   Web Server Instance Directory: This prompt appears only if you are installing the agent for Sun ONE Web Server. Specify the web server instance that this agent will protect. Enter the full path to the directory where the web server instance is located. Example: / *web_server_root* / https-*mycomputer.example*.com

   Configuration Directory: Specify the directory where the `httpd.conf` file is located. This field appears only when you are installing the agent for IBM HTTP server.

Apache Binary Directory: Select the directory where the Apache binary, that is, `httpd` binary is installed. This field appears only when you are installing the agent for Apache Server.

Lotus Domino Data directory: Enter the full path to the directory where the Domino data is located. The default data directory is `/local/notesdata`. This field is available only if you are installing the policy agent for Lotus Domino.

Web Server Port: Enter the port number for the web server that will be protected by the agent.

Web Server Protocol: If the web server has been configured for SSL, choose HTTPS; otherwise choose HTTP.

Agent Deployment URI: Enter a Universal Resource Identifier (URI), which will be used to access the agent. The default value is `/amagent`.

NOTE The agent uses the value of the `com.sun.am.policy.agents.agenturiprefix` property to support some essential functions such as notification and post-data preservation. Agent URI prefix is a configurable subset of Agent Deployment URI. It is important to set a valid URL for this property. Its value should be `http://`*host.domain:port/agent_deployment_uri* where *host*, *domain* and *port* are FQDN and port number of the web server where the agent is installed and *agent_deployment_uri* is the URI where the web server will look for agent's related HTML pages. Its default value is `amagent`.

SSL Ready: The installation program displays this option only when you are installing the Apache web server agent. Select this option if the Apache web server you are using has support for SSL. Your Apache web server is considered SSL ready if it has support for `mod_ssl` and its sources have been compiled using EAPI rule.

To find out if your Apache web server has been compiled with the EAPI flag, go to the `bin` directory of the Apache web server and type the command:

```
# ./httpd -V
```

You can see various flags that the Apache web server was compiled with. If the flag `-D EAPI` is displayed in this list, it indicates that your Apache Web Server is SSL ready. However, if you do not see this flag, it does not necessarily indicate that the Web Server does not have support for `mod_ssl`.

The supported configuration for Apache web server are:

- Apache web server without `mod_ssl` support

- Apache web server with `mod_ssl` and EAPI flag enabled.

NOTE Apache web server with `mod_ssl` support and EAPI flag disabled configuration is not supported by Sun ONE Identity Server policy agents.

9. When you have entered all the information correctly, click Next.
10. Enter information about the web server that runs Sun ONE Identity Server. The policy agent will connect to this server.

    Primary Server Host: Enter the FQDN of the system where the primary web server that runs Sun ONE Identity Server is installed. For example, *myserver.eng.example*.com.

    Primary Server Port: Enter the port number for the web server that runs Sun ONE Identity Server.

    Primary Server Protocol: If the web server that runs Sun ONE Identity Server is SSL-enabled, select HTTPS; otherwise select HTTP.

    Primary Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

    Primary Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

    Failover Server Host: Enter the FQDN for the secondary web server that will run Sun ONE Identity Server if the primary web server becomes unavailable. If no failover server host exists, then leave this field blank.

    Failover Server Port: Enter the port number of the secondary web server that runs Sun ONE Identity Server. If no failover server host exists, then leave this field blank.

    Failover Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`. If no failover server host exists, then leave this field blank.

    Failover Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`. If no failover server host exists, then leave this field blank.

Agent Identity Server Shared Secret: Enter the password for the Identity Server internal LDAP authentication user (`amldapuser`).

Re-enter Shared secret: Re-enter the password for the Identity Server internal LDAP authentication user.

CDSSO Enabled: Check this box if you want to enable CDSSO.

11. When all the information is entered correctly, click Next.
12. Review the Installation Summary to be sure that the information you've entered is correct. Note that it displays the CDCServlet URL if you have checked the CDSSO Enabled box in the previous panel.

    If you want to make changes, click Back. If all the information is correct, click Next.
13. In the Ready to Install panel, click Install Now.
14. When the installation is complete, you can click Details to view details about the installation, or click Exit to end the installation program.
15. If you are installing the agent for Sun ONE Web Server or Apache web server, you must restart the web server for the installation to be complete.

NOTE If you are installing the policy agent for Lotus Domino 5, you must configure the Domino DSAPI filter after installation. See the section "Web or Web Proxy Server Running in SSL Mode" on page 47 for detailed steps.

Installing the Policy Agent for a Web Proxy Server

Use these instructions to install the policy agent for Sun ONE Web Proxy Server 3.6 (in reverse proxy mode) using the GUI on the Solaris 8 operating system.

You must have root permissions when you run the agent installation program.

1. Unpack the product binaries using the following command:

   ```
   # gunzip -dc agent_SunOS_proxy.tar.gz | tar -xvof -
   ```
2. Run the `setup` program. You will find the program in the directory where you untarred the binaries. At the command line, enter the following:

   ```
   # ./setup
   ```
3. In the Welcome page, click Next.
4. Read the License Agreement. Click Yes to agree to the license terms.

5. To search for the directory where you would like to install the agent, click Browse. To accept the default, click Next.

6. When prompted, provide the following information about the web proxy server where this agent will be installed:

   Host Name: Enter the FQDN of the system where the remote web server is installed. For example, *mycomputer.example*.com.

   Proxy Server Instance Directory: Enter the full path to the directory where the Sun ONE Web Proxy Server instance is located. For example:

   proxy_server_root_dir / proxy-*mycomputer*-proxy

   Proxy Server Port: Enter the port number for the Proxy server instance.

   Proxy Server Protocol: If the proxy server has been configured for SSL, choose HTTPS; otherwise choose HTTP.

   Agent Deployment URI: Enter a directory name. The default URI is `/amagent`.

7. When all the information is entered correctly, click Next.

8. When prompted, provide the following information about the web server that runs Sun ONE Identity Server.

   Primary Server Host: Enter the FQDN of the system where the primary web server that runs Sun ONE Identity Server is installed. For example, *myserver.example*.com.

   Primary Server Port: Enter the port number for the web server that runs Sun ONE Identity Server.

   Primary Server Protocol: If the web server that runs Sun ONE Identity Server is SSL-enabled, select HTTPS; otherwise select HTTP.

   Primary Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

   Primary Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

   Failover Server Host: Enter the FQDN for the secondary web server that will run Sun ONE Identity Server if the primary web server becomes unavailable. If no failover host exists, then leave this field blank.

Failover Server Port: Enter the port number of the secondary web server that runs Sun ONE Identity Server. If no failover host exists, then leave this field blank.

Failover Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

Failover Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

Agent Identity Server Shared Secret: Enter the password for the Identity Server internal LDAP authentication user.

Re-enter Shared secret: Re-enter the password for the Identity Server internal LDAP authentication user.

CDSSO Enabled: Check this box if you want to enable CDSSO.

9. When all the information is entered correctly, click Next.
10. Review the Installation Summary to be sure that the information you've entered is correct. Note that it displays the CDCServlet URL if you have checked the CDSSO Enabled box in the previous panel. If you want to make changes, click Back. If all the information is correct, click Next.
11. In the Ready to Install page, click Install Now.
12. When the installation is complete, you can click Details to view details about the installation, or click Exit to end the Installation program.
13. Restart the Proxy Server.

Installing From the Command Line

The following sections describe how to use the CLI of the installation program to install the web policy agents.

Installing the Policy Agent for a Web Server

1. In the directory where you unpacked the binaries, at the command line, enter the following:

```
# setup -nodisplay
```

2. Set your JAVAHOME environment variable to a JDK version 1.3.1_04 or higher. The installation requires that you set up your JAVAHOME variable correctly. However, in case you have incorrectly set the JAVAHOME variable, the `setup` script will prompt you for supplying the correct JAVAHOME value:

   ```
   Please enter JAVAHOME path to pick up java:
   ```

3. Type the full path to the directory where JDK is located and press Enter.

4. When prompted, provide the following information:

   Have you read, and do you accept, all of the terms of the preceding Software License Agreement? Enter yes.

   Install Sun ONE Identity Server Agent in this directory: Enter the full path to the directory in which you want to install the policy agent.

5. Provide the following information about the web server this agent will protect:

   - Host Name
   - Port
   - Web Server Instance Directory (if you are installing the policy agent for Lotus Domino, IBM HTTP server, or Apache Server, the installation program will prompt for the Lotus Domino Data Directory, the Configuration Directory or the Apache Binary Directory as appropriate.)
   - Web Server Protocol
   - Agent Deployment URI
   - SSL Ready (only for Apache agent)

   For more information on each of these items, see “Installing the Policy Agent for a Web Server.”

6. Provide the following information about the web server that runs Sun ONE Identity Server:

   - Primary Server Host
   - Primary Server Port
   - Primary Server Protocol
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host

- Failover Server Port
- Failover Server Deployment URI
- Failover Console Deployment URI
- Agent-Identity Server Shared secret
- Re-enter Shared secret
- CDSSO Enabled

For more information on each of these items, see "Installing the Policy Agent for a Web Server."

The following text is displayed:

```
Ready to Install

1. Install Now
2. Start Over
3. Exit Installation
```

7. When prompted, **What would you like to do?**, enter 1 to start the installation.

   The following text is displayed:

```
Product                           Result     More Information
1.  Sun ONE Identity Server Agent  Installed  Available
2.  Done
```

8. To see log information, enter 1. To exit the installation program, enter 2.

Installing the Policy Agent for a Web Proxy Server

1. In the directory where you unpacked the binaries, at the command line, enter the following:

   ```
   # setup -nodisplay
   ```

2. When prompted, provide the following information:

   Have you read, and do you accept, all of the terms of the preceding Software License Agreement? Enter yes.

   Install Sun ONE Identity Server Agent in this directory: Enter the full path to the directory in which you want to install the policy agent.

3. Provide the following information about the Web Server this agent will protect:

   - Host Name
   - Proxy Server Instance Directory
   - Proxy Server Port
   - Proxy Server Protocol
   - Agent Deployment URI

   For more information on each of these items, see "Installing the Policy Agent for a Web Server."

4. Provide the following information about the proxy web server that runs Sun ONE Identity Server:

   - Primary Server Host
   - Primary Server Port
   - Primary Server Protocol
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host
   - Failover Server Port
   - Failover Server Deployment URI
   - Failover Console Deployment URI
   - Agent-Identity Server Shared secret
   - Re-enter Shared secret

- CDSSO Enabled

For more information on each of these items, see "Installing the Policy Agent for a Web Proxy Server."

The following text is displayed:

```
Ready to Install

1. Install Now
2. Start Over
3. Exit Installation
```

5. When prompted, **What would you like to do?**, enter 1 to start the installation.

   The following text is displayed:

```
Product                          Result     More Information
1.  Sun ONE Identity Server Agent  Installed  Available
2.  Done
```

6. To see log information, enter 1. To exit the installation program, enter 2.
7. Restart the Proxy Server.

Post-installation Tasks

The following sections present tasks you need to perform after the installation of some of the agents.

Configuring the Domino DSAPI Filter

Use the following procedure to configure DSAPI filter if you are installing the policy agent for IBM Lotus Domino:

1. In Lotus Domino Administrator, choose Administrator Tab > Server > All Server Documents.

2. From the listed servers, select the required server.
3. Click Internet Protocols > HTTP tab.
4. At the DSAPI Filter File Names field, enter the file names as follows:
   - For IBM Lotus Domino 5.0.11, enter *Agent_Install_Dir*/SUNWam/Agents/Domino/lib/libamdomino.so
   - For IBM Lotus Domino 6.5, enter *Agent_Install_Dir*/SUNWam/agents/domino6/lib/libamdomino6.so
5. Click the Save and Close button to save the changes.
6. Open Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   load http
   ```

Setting File Ownership and Permissions

For the agents for IBM Lotus Domino to work properly, make sure that the user that Domino server is running as has 'read' permission to the following files:

- *Agent_Install_Dir*/SUNWam/agents/domino6/lib/libamdomino6.so
- *Agent_Install_Dir*/domino6/config/_opt_lotus_notes_notesdata/AMAgent.properties
- /var/tmp/debug/_opt_lotus_notes_notesdata/amAgent

To set the required permission, you can use the commands as shown in the following examples:

```
chown notes:notes Agent_Install_Dir/SUNWam/agents/domino6/lib/libamdomino6.so
chmown notes:notes
/Agent_Install_Dir/SUNWam/domino6/config/_opt_lotus_notes_notesdata/AMAgent.prop
erties
chown notes:notes  (+w) /var/tmp/debug/_opt_lotus_notes_notesdata/amAgent
```

In the above examples, notes is the default user created during IBM Lotus Domino installation.

Additionally, if Sun ONE Identity Server is running with SSL, the files `cert7.db` and `key3.db` must also allow 'read' access to the user the Domino Server is running as. These files are available in the directory specified by the property `com.sun.am.sslCertDir` in the `AMAgent.properties` file.

For example, if the property is set as `com.sun.am.sslCertDir = /opt/my-agents-dir`, ensure that `/opt/my-agents-dir/{cert7.db,key3.db}` has the necessary permissions by using the following command:

```
chown notes:notes /opt/my-agents-dir/cert7.db /opt/my-agents-dir/key3.db
```

Configuring the Agent for Multiple Web Server Instances

To configure an agent for multiple web server instances on a single computer, use the GUI or command-line version of the agent installation program to install the first agent. After the first agent is installed, you can then configure the agent for multiple web server instances using the `config` script. This script must be run from the command line as described in the next section.

In this release, you cannot install more than one type of agents on the same machine. For example, you cannot install an Apache agent and a Sun ONE Web Server agent on the same machine.

Configuring the Agent for Multiple Web Server Instances on the Same Computer

Once you have installed an agent on a system, you can configure it for multiple instances of the web server on that system using the `config` script that is copied into the system during the agent installation. The two scripts, `config` and `unconfig`, are located in the following directory:

Agent_Install_Dir`/SUNWam/agents/`*WS_TYPE*`/bin`

WS_TYPE can be es6, proxy, apache or domino depending on which web server the agent is protecting.

NOTE On HP-UX 11.11, you must use the scripts `config_es6` and `unconfig_es6` to configure and unconfigure the agent for multiple web server instances. These scripts are available in the following directory:

Agent_Install_Dir`/agents/`*WS_TYPE*`/bin`

1. To configure the agent for additional web server instances on a system, run the `config` script from the `bin` directory using the following command:

   ```
   # ./config
   ```

2. Follow the prompts to configure additional web server instances. For information on each of the prompts, see "Installing the Policy Agent for a Web Server." In general, information needs to be entered for both the protected web server instance and the Sun ONE Identity Server server(s). The following text shows an example.

   If you are installing the agent for Lotus Domino, the following example will show the Lotus Domino Data Directory in the place of the Web Server Instance Directory.

```
# ./config
Enter the Web Server Instance Directory: [/web_server_root/https-server_instance]
Enter the Local Hostname: [mycomputer.eng.example.com]
Enter the Agent Web Server Port: [80]
Select Agent Web Server Protocol: [1] http [2] https-->[1]
Enter the Agent Deployment URI: [/amagent]
Select Identity Server Protocol: [1] http [2] https --> [1]
Enter the Identity Server Hostname: [mycomputer.eng.example.com]
Enter the Identity Server Port: [58080]
Enter the Identity Server Deployment URI [/amserver]
Enter the Identity Server's Console Deployment URI [/amconsole]
Select Failover Identity Server Protocol: [1] http [2] https [3] no failover
--> []
Enter the Failover Identity Server Hostname: []mycomputer.example.com
Enter the Failover Identity Server Port: []
Enter the Identity Server Deployment URI [/amserver]
Enter the Identity Server's Console Deployment URI [/amconsole]
Enter Agent-Identity Server shared secret:
Re-enter Agent-Identity Server shared secret:
Is CDSSO Enabled: [1] yes [2] no --> [2] 1
Configuring webserver ... Webserver version:  6.0
Done
```

NOTE Be sure to use the `unconfig` script to uninstall any agent that was installed using the `config` script—you cannot use the GUI installation program to uninstall agents that were installed from the command line. The GUI uninstallation program must be executed only after unconfiguring all the existing agents using the command-line `unconfig` script.

Deploying the Agent with Multiple Instances of Sun ONE Identity Server

When you have to install multiple instances of Sun ONE Identity Server along with the policy agents, it is recommended that you deploy all the instances of Sun ONE Identity Server first and then install the agents as necessary. If you add another instance of Identity Server after the agents have been installed, you must edit the `magnus.conf` file of the new instance of Identity Server to remove the entries corresponding to the installed agents. Then you can install an agent to protect this instance, if necessary.

Using Secure Sockets Layer (SSL) With an Agent

During installation, if you choose the HTTPS protocol, the agent is automatically configured and ready to communicate over SSL.

NOTE Before proceeding with the following steps, ensure that the web server is configured for SSL.

You should have a solid understanding of SSL concepts and the security certificates required to enable communication over the HTTPS protocol. See the documentation that comes with your web server. If you're using Sun ONE Web Server, you can access the documentation at:

`http://docs.sun.com/source/816-5682-10/esecurty.htm#1011961`

Configuring the IBM HTTP Server

Use the following instructions to configure the IBM HTTP Server to run in SSL mode.

1. Create a new key database using the key management utility (IKEYMAN). For information on creating new key database, see the documentation at:
   http://www-3.ibm.com/software/webservers/httpservers/doc/v1319/9atikeyu.htm#HDRKMU2G
2. Create a self-signed certificate using IKEYMAN. For information on creating a self-signed certificate, see the documentation at:
   http://www-3.ibm.com/software/webservers/httpservers/doc/v1319/9atikeyu.htm#HDRKMU4G
3. Start the Administration Server

   ```
   # /opt/IBMHTTPD/bin/adminctl start
   ```
4. Setup SSL using the IBM Administration Server. For information on setting up SSL, see the documentation at:

   http://www-3.ibm.com/software/webservers/httpservers/doc/v1319/9atstart.htm#ssl

Web or Web Proxy Server Running in SSL Mode

If your web or web proxy server is running in the SSL mode, and your agent is in the notification mode, you must install the root CA certificate of your web or web proxy server onto Identity Server if it is not already installed.

The Agent's Default Trust Behavior

By default, the policy agent installed on a remote web server or proxy server will trust any server certificate presented over SSL by the web server that runs Sun ONE Identity Server; the agent does not check the root Certificate Authority (CA) certificate. If the web server that runs Identity Server is SSL-enabled, and you want the policy agent to perform certificate-checking, you must do the following:

1. Disable the agent's default trust behavior.
2. Install a root CA certificate on the remote web server (where the agent is installed). The root CA certificate must be the same as the one installed on the web server that runs Sun ONE Identity Server service.

Disabling the Agent's Default Trust Behavior

The following property in the `AMAgent.properties` file controls the agent's trust behavior, and by default it is set to `true`:

```
com.sun.am.trustServerCerts=true
```

This means that the agent does not perform certificate checking.

To Disable the Default Behavior

1. Set the following property to `false`:

   ```
   com.sun.am.trustServerCerts=false
   ```

2. Set the directory Cert DB in the file `AMAgent.properties` as shown in the following example:

   ```
   com.sun.am.policy.am.sslCertDir= /opt/SUNWam/servers/alias
   ```

   For Apache agent, set as following:

   ```
   com.sun.am.policy.am.sslCertDir= /etc/apache/cert
   ```

   For IBM HTTP Server, set as following:

   ```
   com.sun.am.policy.am.sslCertDir=/opt/IBMHTTPD/cert
   ```

   For Domino Web Server, set as following:

   ```
   com.sun.am.policy.am.sslCertDir=/opt/domino/cert
   ```

3. Set the Cert DB Prefix, if required.

   In cases where the specified Cert DB directory has multiple certificate databases, the following property must be set to the prefix of the certificate database to be used.

   ```
   com.sun.am.policy.am.certDbPrefix
   ```

   For example, set the property for Sun ONE Web Server as this:

   ```
   com.sun.am.policy.am.certDbPrefix =https-host.domain.com.host-
   ```

Installing the Root CA Certificate on the Remote Web Server

The root CA certificate that you install on the remote web server must be the same one that is installed on the web server that runs Sun ONE Identity Server.

To Install the Root CA Certificate on Sun ONE Web Server

See the instructions for installing a root CA certificate in the documentation that comes with the web server. Generally, you install a root CA certificate through the web server's Administration console.

You can access the documentation for Sun ONE Web Server 6.0 on the Internet at the following URL:

http://docs.sun.com/source/816-5682-10/esecurty.htm#1011961

To Install the Root CA Certificate on Apache 1.3.27

You can use the `certutil` program to install the root CA certificate on Apache 1.3.27.

1. In C shell, at the command line, enter the following commands (assuming `/etc/apache` is the directory where the apache configuration file is located):

   ```
   # cd /etc/apache/cert
   ```

   ```
   # setenv LD_LIBRARY_PATH
   /Agent_Install_Dir/SUNWam/agents/apache/lib:/Agent_Install_Dir/SUNWam/agents/lib:/usr/lib/mps
   ```

2. Create the necessary certificate database if you have not already done so.

   ```
   # /Agent_Install_Dir/SUNWam/agents/apache/cert/certutil -N -d .
   ```

3. Install root CA certificate.

   ```
   # /Agent_Install_Dir/SUNWam/agents/apache/cert/certutil -A -n cert-name
   -t "C,C,C" -d cert-dir -i cert-file
   ```

   In the commands above, the variables represent the following:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located.

- *cert-file* is the base-64 encoded root CA certificate file.

For more information on the `certutil` utility, enter `certutil -H` for online Help.

4. To verify that the certificate is properly installed, at the command line, enter the following:

```
# ./certutil -L -d .
```

Trust database information will be displayed including the name of the root CA certificate you installed. For example:

```
Certificate Name                                Trust Attrubutes

cert-name                                       C,C,C

p    Valid peer
P    Trusted peer (implies c)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

To Install the Root CA Certificate on Web Proxy Server

You can use the `certutil` program to install the root CA Certificate on Proxy Server.

1. In C shell, at the command line, enter the following commands:

```
# mkdir Proxy_Server_Instance_Dir/cert
```

```
# cd Proxy_Server_Instance_Dir/cert
```

```
# setenv LD_LIBRARY_PATH
/Agent_Install_Dir/SUNWam/agents/proxy/lib:/Agent_Install_Dir/SUNWam/agent
s/lib:/usr/lib/mps
```

2. Create the necessary certificate database if you have not already done so.

3. Install root CA certificate.

   # */Agent_Install_Dir*/SUNWam/agents/proxy/cert/certutil -N -d .

   # */Agent_Install_Dir*/SUNWam/agents/proxy/cert/certutil -A -n cert-name -t "C,C,C" -d cert-dir -i cert-file

   In the commands above, the variables represent the following:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located.
   - *cert-file* is the base-64 encoded root CA certificate file.

   For more information on the `certutil` utility, enter `certutil -H` for online Help.

To Install the Root CA Certificate on IBM HTTP Server 1.3.19

You can use the `certutil` program to install the root CA certificate on IBM HTTP Server 1.3.19.

1. In C shell, at the command line, enter the following commands:

   # `mkdir` *server_instance_dir*`/cert`

   # `setenv LD_LIBRARY_PATH` */Agent_Install_Dir*`/SUNWam/agents/ibmhttp/lib:`*/Agent_Install_Dir*`/SUNWam/agents/lib`

2. Create the necessary certificate database if you have not already done so.

   # */Agent_Install_Dir*/SUNWam/agents/ibmhttp/cert/certutil -N -d .

NOTE This will create the following three files in the directory *server_instance_dir*/cert:

- secmod.db
- key3.db
- cert7.db

Make sure that the UNIX user that the IBM HTTP server runs as (for example nobody), has read permissions on these three files.

3. Install root CA certificate.

   # */Agent_Install_Dir*/SUNWam/agents/ibmhttp/cert/certutil -A -n cert-name -t "C,C,C" -d cert-dir -i cert-file

   In the command above, the variables represent the following:

- *cert-name* can be any name for this root CA certificate.
- *cert-dir* is the directory where the certificate and key stores are located.
- *cert-file* is the base-64 encoded root CA certificate file.

For more information on the `certutil` utility, enter `certutil -H` for online Help.

4. To verify that the certificate is properly installed, at the command line, enter the following:

   ```
   # ./certutil -L -d .
   ```

   Trust database information will be displayed including the name of the root CA certificate you installed. See the following example.

```
Certificate Name                                Trust Attrubutes

cert-name                                       C,C,C

p    Valid peer
P    Trusted peer (implies c)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

To Install the CA Certificate on Domino Web Server

See the instructions for installing a CA Certificate in the documentation that comes with the web server. Generally, this is done through the web server's Administration console.

1. Go to the following directory:

   Agent_Install_Dir`/Agents/domino/utils`

2. Add the same certificate that is installed on the web server that runs Identity Server services into the existing certificate database. At the command line, enter the following command:

   ```
   certutil -A -n cert-name -t "C,C,C" -d cert-dir -i cert-file
   ```

3. In the command above, the variables represent the following:
   - *cert-name* can be any name for this certificate.
   - *cert-dir* is the directory where the certificate and key stores are located. The location is:

     Agent_Install_Dir/Agents/domino/cert
   - *cert-file* is the base-64 encoded certificate file.

   For more information on certutil, type certutil -H
4. Restart Domino Web Server.

Setting the REMOTE_USER Server Variable

The REMOTE_USER server environment variable can be set to a Identity Server authenticated user or anonymous user. By setting this variable to a specific user, the user becomes available to web applications (such as a CGI, servlet, or ASP program). This feature makes it possible to personalize the content of displayed HTML pages to specific users.

To enable the REMOTE_USER setting for globally not-enforced URLs as specified in the AMAgent.properties file (these are URLs that can be accessed by unauthenticated users), you must set the following property in the AMAgent.properties file to TRUE (by default, this value is set to FALSE):

```
com.sun.am.policy.agents.anonRemoteUserEnabled=TRUE
```

When you set this property value to TRUE, the value of REMOTE_USER will be set to the value contained in the following property in the AMAgent.properties file (by default, this value is set to anonymous):

```
com.sun.am.policy.agents.unauthenticatedUser=anonymous
```

NOTE This feature is not available for the Sun ONE Web Proxy Server agent.

Validating Client IP Addresses

This feature can be used to enhance security by preventing the stealing or *hijacking* of SSOTokens.

The AMAgent.properties file contains a property titled com.sun.am.policy.agents.client_ip_validation_enable, which by default is set to false.

If you set this property value to true, client IP address validation will be enabled for each incoming request that contains an SSO token. If the IP address from which request was generated does not match the IP address issued for the SSO token, the request will be denied. This is essentially the same as enforcing a deny policy.

This feature should not be used, however, if the client browser uses a web proxy or if there is a load-balancing application somewhere between the client browser and the agent-protected web server. In such cases, the IP address appearing in the request will not reflect the real IP address on which the client browser runs.

POST Data Preservation

POST data preservation is supported on the Sun ONE Web Server 6.0 SP*x* agent. Users can preserve POST data, which are submitted to web servers through HTML forms before users login to Sun ONE Identity Server. Presumably, the HTML page containing the form should be in the not-enforced list. By default, this feature is set off.

This feature is configurable through two properties in AMAgent.properties file. To turn off this feature, use the following AMAgent.properties file property and change the value of the property from true to false:

```
com.sun.am.policy.agents.is_postdatapreserve_enabled = true
```

```
com.sun.am.policy.agents.postcacheentrylifetime = 10
```

The second property decides how long any POST data can stay valid in the web server cache. After the specified interval, a reaper thread will wake up and clean up any POST cache entries that have lived beyond the specified life time. The following property helps the administrator to configure this time interval. By default, this property is set to 10 minutes.

NOTE This feature is available only on the agent for Sun ONE Web Server 6.0 SPx.

Shared Secret Encryption Utility

The policy agent stores the shared secret in the `AMAgent.properties` file. By default, this password is the Identity Server internal LDAP authentication user password. This can be changed on the server side by editing the `AMConfig.properties` file.

The property `com.sun.am.policy.am.password` in the `AMAgent.properties` file is set with the encrypted shared secret while installing the agent.

To reset or change the shared secret, you can use the following utility and set the value in the property.

1. Go to the following directory:

   Agent_Install_Dir`/bin`

2. Execute the following script from the command line:

   ```
   # ./crypt_util shared_secret
   ```

3. Cut and paste the output from Step 2 in the property:

   `com.sun.am.policy.am.password`

4. Restart the web server and try accessing any resource protected by the agent.

Uninstalling a Policy Agent

The following sections provide steps for uninstalling the agent. Note the following:

- Be sure to use the `unconfig` script to uninstall any agent that was installed using the `config` script.The uninstallation program must be executed only after unconfiguring all the existing agents using the command-line `unconfig` script.
- If you want to uninstall the web server for some reason, make sure that you uninstall the agent before you uninstall the web server.

Unconfiguring a Policy Agent

To remove an agent that was configured using the `config` script, use the script `unconfig`. The `unconfig` script is located in the following directory:

Agent_Install_Dir`/SUNWam/agents/es6/bin`

WS_TYPE can be es6, proxy, apache or domino depending on which web server the agent is protecting.

On HP-UX 11.11, you must use the script `unconfig_es6` instead of the `unconfig` script. You can find the script in the following directory:

Agent_Install_Dir`/agents/es6/bin`

Here is an example run of the `unconfig` script.

```
# ./unconfig /web_server_root/https-server_instance
Unconfiguring webserver ...
done.
```

NOTE If you are removing the agent for Lotus Domino 5.0.10, you should specify the path to the Domino data directory instead of `https-`*server_instance*. Additionally, you should remove the DSAPI filter file for the required Domino server using the Domino Administrative Client and then restart the Domino Web Server.

Before Uninstalling the Policy Agent for Lotus Domino

Before you uninstall the policy agent for Lotus Domino 5.0.11 or 6.0.1, you should perform the following steps on the Lotus Domino Administrator client from a Windows machine.

1. Launch Lotus Domino Administrator.
2. Choose Administrator Tab > Server > All Server Documents.
3. From the listed servers, select the server you want to uninstall.
4. Click Internet Protocols > HTTP tab.
5. Remove the DSAPI filter file name specified for the Lotus Domino agent.
6. Click the Save and Close button to save the changes.

7. Open the Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   ```

   ```
   load http
   ```

Uninstalling Using the GUI

To uninstall an agent, you must run the uninstallation program. Follow the steps below:

1. In the directory where the agent is installed, at the command line, enter the following command:

   ```
   # ./uninstall_agent
   ```

   On HP-UX 11.11, use the following command:

   ```
   java uninstall_Sun_ONE_Identity_Server_Policy_Agent
   ```

2. Click Next on Welcome panel.
3. Click Uninstall Now on Ready to Uninstall panel.
4. Click Close after uninstallation is complete.
5. Restart the web server.

Uninstalling From the Command-Line

1. From the directory where the agent is installed, enter the following command at the command line:

   ```
   # ./uninstall_agent -nodisplay
   ```

   The uninstallation program detects the agent that was previously installed using the `setup` program. Enter `1` to uninstall the agent.

   The following text is displayed:

```
Ready to Uninstall

1. Uninstall Now
2. Start Over
3. Exit Uninstallation
```

2. When prompted, **What next?** enter `1` to begin uninstallation.

   The following text is displayed:

```
Product                            Result  More Information
1.  Sun ONE Identity Server Agent  Full    Available
2.  Done
```

3. To see log information, enter `1`. To exit the uninstallation program, enter `2`.
4. Restart the web server.

Troubleshooting

Cannot install the agent after a previous installation is removed

The following is an example message that is displayed when you run the agent installation program:

```
"Sun ONE Identity Server Policy Agent 2.1 for Sun ONE Web Server 6.0 SPx is
installed.  Please refer to installation manual to configure this agent for
another web server instance. Or uninstall it before installing another
agent."
```

Possible Causes:

- You might have an existing installation of the agent.

- You might have a previously-installed agent and did not use the agent's uninstallation program to uninstall the agent.
- The installation program's `productregistry` file may be corrupted.

Solution:

- Check that you have uninstalled any existing installation of the agent.
- The `productregistry` file may be corrupted if there is no existing installation of the agent. This file is used by the installation program to track installed products. It is found in `/var/sadm/install` directory.

NOTE Make a backup copy of this file before you make changes.

Remove the agent entry in this file. This entry starts with the following lines:

```
<compid>SUNWamcom
   <compversion>2.1
      <uniquename>SUNWamcom</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>Agent uninstall script
   <compversion>2.1
      <uniquename>Agent uninstall script</uniquename>
      <vendor>Sun Microsystems, Inc.</vendor>
      ......
</compid>
<compid>Agent installer resource bundle
   <compversion>2.1
      <uniquename>Agent installer resource bundle</uniquename>
      <vendor>Sun Microsystems, Inc.</vendor>
      ......
</compid>
<compid>Agent Common Core and SDK
   <compversion>2.1
      <uniquename>Agent Common Core and SDK</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>SUNWames6
   <compversion>2.1
      <uniquename>SUNWames6</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>Agent for ...
   <compversion>2.1
      <uniquename>Agent for ...</uniquename>
      <vendor></vendor>
```

```
      ......
</compid>
<compid>Sun ONE Identity Server Policy Agent
   <compversion>2.1
      <uniquename>Sun ONE Identity Server Policy Agent</uniquename>
</compid>
```

The uninstallation program does not remove entries from the agent's web server if there is another instance of web server configured using the Configuration script.
Solution:

You should remove all the instances of the agent using the `unconfig` script before running the uninstallation program.

On Solaris 8, the agent's performance may suffer when multiple threads are running.
Solution:

On Solaris 8:

1. Apply the latest Solaris 8 cluster patch (Patch Cluster Date Jan / 24 / 03 or later).
2. Download and install patch 111308-03 or later (showrev -p | grep 111308) from http://sunsolve.central.sun.com.
3. Reboot the machine after applying the patches (this is recommended but not mandatory).

For agents running on Sun ONE Web Server 6.0:

1. In your web server instance directory, open the start script.
2. Before the first LD_LIBRARY_PATH line, add the following two lines

   ```
   LD_PRELOAD=libmtmalloc.so.1
   ```

   ```
   export LD_PRELOAD
   ```
3. To make sure the patches are installed correctly, try this. It should produce a line of output.

   ```
   showrev -p | grep 111308
   ```
4. To verify that `libmtmalloc` is being used by the web server, do the following.

   ```
   ps -ef |grep httpd | grep nobody
   ```

5. Take the pid from the previous step and do

   `pldd pid of the process | grep mtmalloc`

   The output should show `/usr/lib/libmtmalloc.so.1`.

The browser goes into a loop for a minute or so before displaying an access-denied page when a user tries to access a resource for which a policy with a time condition has been set and the time on the agent host and the Identity Server host are not in sync.

Solution:

Login as `root` and run the command `rdate` *hostname* to synchronize the time on both the hosts.

Known Problems

After installing the Solaris 2.8 Patch # 109234-09, Apache Server does not startup correctly or hangs

This can happen if your system does not have `JServ` installed at the time of installing the patch. In order to correct this problem, edit your `httpd.conf` file located under `/etc/apache/` directory and comment the following line:

`LoadModule jserv_module /usr/apache/libexec/mod_jserv.so` and include `/etc/apache/jserv.conf`

Error message displayed during startup

Apache server displays the following error message during startup after the agent is installed:

```
Syntax error on line 1 of
/etc/opt/SUNWam/agents/apache/config/_usr_local_apache_conf/dsame.conf:

Invalid command 'LoadModule', perhaps mis-spelled or defined by a module not
included in the server configuration
./apachectl start: httpd could not be started
```

This indicates that the Apache server does not have `mod_so` enabled and consequently does not support dynamic shared objects. To enable `mod_so` support, refer Apache Server documentation at http://httpd.apache.org/.

Policies not working with Sun ONE Web Proxy Server agent.

The resource names for setting policies for the reverse proxy agent should be the URLs qualified as “URL prefix (from client)” in the Sun ONE Web Proxy Server 3.6 administrative console.

Agents are not effective anymore after the modification of Sun ONE Web Server configuration using the administrative console.

Modifications to the configuration of the web server’s configuration files (`obj.conf`, `magnus.conf`) may be overwritten by the server, which would disable the agent completely. To ensure this does not occur, after installing the agent, go to the Web Server’s administration console, and press the Apply Button. You could either load the configuration or click on Apply Changes that would restart the server. If you do get into this problem, go to the `bin` directory where `unconfig` script is located. Unconfigure the agent and reconfigure it for this instance of the web server. After doing this, go to the Administration console of the web server and click the Apply button to save the configuration changes made by the agent.

Chapter 3

Policy Agents on Microsoft Windows

This chapter explains how to install and configure Web Policy Agents available for the web servers running on Microsoft Windows operating system.

Topics include:

Before You Begin

Be sure that your are familiar with the concepts presented in Chapter 1, “Read This First.” The chapter includes brief but important information on the following topics:

- Supported Servers
- Web Policy Agents
- Java Runtime Environment 1.3.1 or Higher
- Remote Web Servers
- Configuring the Agent for Multiple Web Server Instances on the Same Computer
- Providing Failover Protection for Agents
- Updating the Agent Cache
- Not-Enforced URL List
- Not-Enforced IP Address List
- Enforcing Authentication Only
- Forwarding LDAP User Attributes via HTTP Headers
- The Agent Properties File
- Setting the Fully Qualified Domain Name
- Configuring CDSSO

Overview of Policy Agents for Microsoft Windows

The Sun ONE Identity Server policy agents that support Microsoft Windows have some variety and the web servers they are deployed on can vary a great deal. For example, the architecture of Microsoft IIS web servers is quite unique, which can affect how you prepare for the installation of the policy agent. For more information, see “Preparing for Agent Installation on Microsoft IIS Web Servers” on page 66.

Furthermore, some variety exists among the policy agents for Microsoft Windows in how they are installed and configured. The installation of the agents for Microsoft Windows can be categorized into two distinct types. Therefore, for purposes of written organization these agents have been divided in this chapter into the two following groups.

- Installation Type I
- Installation Type II

The agent installation program can be run in two modes: the graphical user interface (GUI) mode and the command-line interface (CLI) mode. The different installation types apply to the GUI mode. The CLI mode applies to all agents regardless of installation type.

Supported Servers for Microsoft Windows

The following table shows the installation type for each Sun ONE Identity Server policy agent that supports Microsoft Windows. The table also specifies which platforms each agent supports.

Table 3-1 Supported Servers for Microsoft Windows

Agent for	Supported Platform
Installation Type I	
Sun ONE Web Server 6.0 SP*x*	Microsoft Windows 2000
IBM Lotus Domino 6.0.1	Microsoft Windows 2000
IBM Lotus Domino 5.0.11	Microsoft Windows 2000
Microsoft IIS 5.0	Microsoft Windows 2000
Sun ONE Web Server 6.1	Microsoft Windows 2000
SAP Internet Transaction Server 2.0	Microsoft Windows 2000 Advanced Server
Sun Java System Web Server 6.1	Microsoft Windows Server 2003 EE
IBM Lotus Domino 6.5	Microsoft Windows 2000
	Microsoft Windows Server 2003 EE
Installation Type II	
Microsoft IIS 6.0	Microsoft Windows Server 2003 EE
Apache 2.0.50	Microsoft Windows Server 2003 EE

The Agent Installation Types

All the Installation Type I agents share the same installation steps. The agents for Domino 5 and 6 require additional configuration of the Domino server.

Both the Installation Type II agents, Microsoft IIS 6.0 and Apache 2.0.50, also share the same installation steps. However, after the installation steps are completed, both of these agents require configuration. The configuration steps are divided into two sets of steps. The three sets of procedures for installation and configuration of the Installation Type II agents are as follows:

1. Installing the agent
2. Creating the agent configuration file for the web site that is to be protected by the agent
3. Configuring the agent for that web site

Preparing for Agent Installation on Microsoft IIS Web Servers

The policy agent for Microsoft IIS 5.0 enforces policy on URL access for Microsoft's IIS web server. The agent is an IIS ISAPI filter installed at the IIS web service level that will enforce policy on all IIS web sites. Technical considerations prevent the agent from being installed at the web site level.

Prior to installation, be sure that the entry for the system where the agent will be installed has a domain name set. If the web server that runs Sun ONE Identity Server 6.0 is running on a separate system, make sure the server is also in the DNS query list.

NOTE The policy agent for Microsoft IIS 6.0 is an ISAPI application. It is deployed as a wildcard application mapping to a web site. This implies that the agent for Microsoft IIS 6.0 when deployed for a particular web site intercepts every request for accessing the resources on that web site. It does authentication and policy evaluation. If all the conditions are met the agent allows access to the resource.

One new feature in the agent for Microsoft IIS 6.0 is that multiple agents can be configured on the same machine. This was not possible in the policy agent for Microsoft IIS 5.0. Using this new feature, the policy agent can be configured for multiple web sites on multiple application pools. However, the agent cannot currently be configured for multiple web sites on the same application pool.

Installing and Configuring the Installation Type I Agents

This section explains how to install most policy agents supported on Microsoft Windows. For a list and explanation of the Installation Type I and Installation Type II agents, see "Overview of Policy Agents for Microsoft Windows" on page 64. Of the Installation Type I agents, only the agents for Lotus Domino 5.0 and 6.0 require configuration. Therefore, this section provides the configuration steps for them only. This section does not explain how to install the agents for Microsoft IIS 6.0 and Apache 2.0.50. For instructions on installing the agents for these two web servers see "Installing the Installation Type II Agents" on page 75.

The following steps help you to install the Installation Type I policy agents in the GUI mode. To install an agent in command-line mode, see "Installing Any Agent from the Command Line" on page 72.

You must have administrator privileges to run the installation program.

1. Unzip the product binaries.

NOTE On Microsoft Windows 2003, the zip file is not automatically unpacked. Therefore, after you download the agents zip file, be sure to extract the zip file to a directory first and then execute `setup.exe`. To extract the zip file, right click on the zip file in the File Manager and select Extract. After extracting to a directory, double click `setup.exe` to execute it.

2. Run the installation program by double-clicking `setup.exe`.
3. In the Welcome window, click Next.
4. Read the License Agreement. Click Yes to accept the license agreement.
5. Select the directory where you want to install the agent.
6. Enter the applicable information about the web server where this agent will be installed in the dialog box.

   The dialog box provides fields for entering the required information. Some of the information required varies depending upon the web server. You are prompted for information in the order shown in the following table.

Table 3-2 Prompted Web Server Information

Web Server	Field	Details
All Supported Web Servers	Host Name	Enter the fully qualified domain name (FQDN) of the system where the agent web server is installed. For example, `mycomputer.example.com.`
Microsoft IIS Specific	IS Document Root	Enter the document root directory. This directory needs to be accessible by the web server root World Wide Web Publishing Service (w3svc). This field is available only if you are installing the policy agent for Microsoft IIS
Sun ONE Specific	Web Server Instance Directory	Enter the full path to the directory where the Sun ONE Web Server instance is located. This is the web server instance that the agent will protect. For example, */web_server_root*`/https-`*mycomputer.example*`.com`
IBM Lotus Domino Specific	Lotus Domino Data directory	Enter the full path to the directory where the Domino data is located. The default path is `\Lotus\Domino`. This field is applicable only if you are installing the policy agent for Lotus Domino 5.0 or 6.0.
All Supported Web Servers	Server Port	Enter the port number for the web server that will be protected by the agent.

Table 3-2 Prompted Web Server Information

Web Server	Field	Details
	Server Protocol	If your web server has been configured for SSL, then select HTTPS; otherwise select HTTP.
	Agent Deployment URI	Enter a Universal Resource Identifier (URI) on the agent. **NOTE** This URI is used to form the value of the `com.sun.am.policy.agents.agenturiprefix` property in `AMAgent.properties`. The value formed should be valid. The agent uses the value of the `com.sun.am.policy.agents.agenturiprefix` property to support some essential functions such as post data preservation. It is important that the value of this property be valid where the protocol, host, and port are reachable by external users. The following is the default value: *Server_Protocol*`://`*Server_Host*`:`*Server_Port*`/amagent`

7. When all the information is entered correctly, click Next.

8. Provide the following information about the web server that runs Sun ONE Identity Server:

   Primary Server Host: Enter the FQDN of the system where the primary web server that runs Sun ONE Identity Server is installed. For example, `myserver.example.com`.

   Primary Server Port: Enter the port number for the web server that runs Sun ONE Identity Server.

   Primary Server Protocol: If the web server that runs Sun ONE Identity Server is SSL-enabled, select HTTPS; otherwise select HTTP.

   Primary Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

   Primary Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

   Failover Server Host: Enter the FQDN for the secondary web server that will run Sun ONE Identity Server if the primary web server becomes unavailable. If no failover host exists, then leave this field blank.

   Failover Server Port: Enter the port number of the secondary web server that runs Sun ONE Identity Server. If no failover host exists, then leave this field blank.

   Failover Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`. If no failover host exists, then leave this field blank.

   Agent Identity Server Shared Secret: Enter the password for the Identity Server internal LDAP authentication user.

   Re-enter Shared secret: Re-enter the password for the Identity Server internal LDAP authentication user.

   CDSSO Enabled: Check this box if you want to enable the CDSSO feature.

9. If all the information entered is correct, click Next.

10. Review the installation summary to be sure that the information you've entered is correct. Note that it displays the CDCServlet URL if you have checked the CDSSO Enabled box in the previous panel. If you want to make changes, click Back. If all the information is correct, click Next.

11. In the Ready to Install page, click Install Now.

12. When the installation is complete, you can click Details to view details about the installation, or click Close to end the installation program.
13. Restart your computer.

    Restarting your computer is necessary for the agent to work properly. The installation modifies the system path by appending to it the location of the agent libraries. This change takes effect only after your computer is restarted.

Configuration is only necessary for Lotus Domino 5 and 6. You must configure the Domino DSAPI filter as explained in "Configuring the Domino DSAPI Filter" on page 71.

Configuring the Domino DSAPI Filter

The policy agent for Lotus Domino supports Domino versions 5.0 and 6.0. When configuring the DSAPI filter, you must direct the policy agent to use the DSAPI file filter appropriate to the version of Domino you are attempting to protect. Use the following procedure to configure the Domino DSAPI filter.

1. In Lotus Domino Administrator, choose Administrator Tab > Server > All Server Documents.
2. From the listed servers, select the server you want to configure.
3. Click Internet Protocols > HTTP tab.
4. At the DSAPI Filter File names field, enter *Agent_Install_Dir*`\domino\bin\amdomino6.dll`

   If you installing the agent for Lotus Domino 5, enter the DSAPI Filter file name as *Agent_Install_Dir*`\domino\bin\amdomino.dll`.
5. Save the changes and close the window.
6. Open Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   load http
   ```

Configuring Domino DSAPI Filter for Multiple Server Partitions

If you are configuring Domino DSAPI Filter for multiple server partitions, you must:

- Use the same `AMAgent.properties` file for all the supported partitions.

- Configure the filter for each of the server partitions you want to support.

You can configure the filter for the different partitions by performing the following steps:

1. In Lotus Domino Administrator, choose Administrator Tab > Server > All Server Documents.
2. From the listed servers, select the required server.
3. Now go to Internet Protocols > HTTP.
4. At the DSAPI Filter File names field, enter the following:

   Agent_Install_Dir`\domino\bin\amdomino6.dll`

   If you installing the agent for Lotus Domino 5, enter the DSAPI Filter file name as follows:

   Agent_Install_Dir`\domino\bin\amdomino.dll`
5. Save the changes and close the window.
6. Open Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   ```

   ```
   load http
   ```

Installing Any Agent from the Command Line

In addition to the GUI mode of installation, the installation program also offers a command-line interface for installing an agent. The command-line interface applies to all agents, Installation Type I and Installation Type II agents. However, configuration is still necessary for Lotus Domino 5 and 6 amongst the Installation Type I agents and for both the Installation Type II agents.

Use the following steps to install the agent from the command-line.

1. In the directory where you unzipped the binaries, at the command line, enter the following command:

   ```
   # setup.exe -nodisplay
   ```

2. When prompted, provide the following information:

   Have you read, and do you accept, all of the terms of the preceding Software License Agreement?

   Install Sun ONE Identity Server Policy Agent in this directory: Specify the directory where you want the agent to be installed. To accept the default directory that is displayed in brackets, press Enter. Otherwise, enter the full path.

3. When prompted, provide the following information about the web server instance that this agent will protect:

   - Host Name
   - Web Server Instance Directory, Lotus Domino Data Directory or IIS Document Root depending on the web server for which you are installing the agent.
   - Server Port
   - Server Protocol
   - Agent Deployment URI

   For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.

4. When prompted, provide the following information about the web server that runs Sun ONE Identity Server Services:

   - Primary Server Host
   - Primary Server Port
   - Primary Server Protocol
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host
   - Failover Server Port
   - Failover Server Deployment URI
   - Secondary Console Deployment URI
   - Agent-Identity Server Shared Secret
   - Re-enter Shared secret

- CDSSO feature enabled

For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.

5. When displayed, review the summary of installation information you've specified. Press Enter to continue, or enter exclamation mark (!) to exit the program.

   The following text is displayed:

```
Ready to Install

1. Install Now
2. Start Over
3. Exit Installation

What would you like to do
```

6. When prompted, **What would you like to do?**, enter `1` to start the installation.

   The following text is displayed:

```
Product                          Result     More Information
1.  Sun ONE Identity Server Agent  Installed  Available
2.  Done
```

7. To see log information, enter `1`. To exit the installation program, enter `2`.
8. Restart your computer

   Restarting your computer is necessary for the agent to work properly. The installation modifies the system path by appending to it the location of the agent libraries. This change takes effect only after your computer is restarted.

NOTE If the Microsoft IIS 5.0 or the Lotus Domino policy agent was previously installed and uninstalled on your computer, you do not need to reboot the computer if you are installing the same agent in the same directory.

For the Installation Type I Agents, configuration is only necessary for Lotus Domino 5 and 6. You must configure the Domino DSAPI filter as explained in “Configuring the Domino DSAPI Filter” on page 71.

For Installation Type II Agents, configuration is necessary for both agents. For detailed information, see “Configuring the Installation Type II Agents” on page 75.

Installing the Installation Type II Agents

The following steps help you to install the Installation Type II policy agents in the GUI mode. To install an agent in command-line mode, see “Installing Any Agent from the Command Line” on page 72.

To install the policy agent for Microsoft IIS 6.0 or Apache 2.0.50 perform the following steps:

1. Unzip the product binaries.
2. Double click `setup.exe` to run the installation program.
3. In the Welcome window, click Next.
4. Read the License Agreement and click Yes to accept the license agreement.
5. Select the directory where you want to install the agent.

   The default directory is C:`\Sun\Identity_Server\Agents\2.1`. The installation program will install the agent under this directory.
6. If you are given the option, click Create Directory.

   If the directory does not exist a dialog box appears giving you the option to create a directory.
7. Click Install Now. The program installs the agent.
8. Click Yes when the program asks if want to reboot the computer. Once the installation is complete, you must create agent configuration files to configure the agent for web sites. The following section explains the procedure for creating the agent configuration file.

Configuring the Installation Type II Agents

After you have installed an Installation Type II agent, Microsoft IIS 6.0 or Apache 2.0.50, you need to perform the following general procedures:

1. Creating the agent configuration file for the web site that is to be protected by the agent
2. Configuring the agent for that web site

Ensure that you create the agent configuration file before you configure the agent.Refer to the proper section as follows:

- "Creating the Microsoft IIS 6.0 Agent Configuration File" on page 76
- "Creating the Apache 2.0.50 Agent Configuration File" on page 79
- "Configuring the Agent for Microsoft IIS 6.0 for a Web Site" on page 81
- "Configuring the Agent for Apache 2.0.50 for a Web Site" on page 83

Creating the Microsoft IIS 6.0 Agent Configuration File

The agent for Microsoft IIS 6.0 provides a Visual Basic (VB) script to help you create the agent configuration file. When you run it, the VB script prompts for information related to the Web Site Identifier, the agent you are installing, and Sun ONE Identity Server and creates the agent configuration file based on the information. About the Web Site Identifier, Microsoft IIS 6.0 has a unique Identifier associated with every web site on the web server. The Web Site Identifier is displayed when you start the Microsoft Internet Information Services Manager and click Web Sites. The Identifier column indicates the unique identifier associated with every web site.

NOTE When you are deploying the agent on multiple web sites, you must create a unique agent configuration file for each of the web sites. You can use the following steps to create multiple agent configuration files. Only, make sure that you give a unique file name to each of the configuration files.

To create the agent configuration file, perform the below steps.

1. Change to the directory `\Agent_Install_Dir\iis6\bin`. This directory stores the script required to create the agent configuration file.
2. Run the following command:

   ```
   cscript.exe IIS6CreateConfig.vbs defaultConfig
   ```

   where, `IIS6CreateConfig.vbs` is the VB script and `defaultConfig` is the agent configuration file created by this command.

NOTE Make sure that you give a unique name for the configuration file since you will need the same file to unconfigure the agent.

The script prompts for information as it progresses with the creation of the agent configuration file. The following is output from running the command.

```
Microsoft (R) Windows Script Host Version 5.6
    Copyright (C) Microsoft Corporation 1996-2001. All rights reserved

    Copyright c 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms
    ---------------------------------------------------
        Microsoft (TM) Internet Information Server (6.0)
    ---------------------------------------------------
    Enter the Agent Resource File Name [IIS6Resource.en] :

    Fully Qualified Host Name :
    drake1.red.iplanet.com

    Displaying the list of Web Sites and its corresponding Identifiers
    Site Name (Site Id)
    Default Web Site (1)
    Test web site (1285842265)
    Microsoft SharePoint Administration (2)

    Web Site Identifier :
    1
    Protocol [http] :

    Port Number [80] :

    Agent Deployment URI [/amagent] :

    -----------------------------------------------
    Sun Java (TM) Enterprise System Identity Server
    -----------------------------------------------
    Primary Server Host :
    drake2.red.iplanet.com

    Primary Server Protocol [http] :

    Primary Server Port [80] :

    Primary Server Deployment URI [/amserver] :

    Primary Server Console URI [/amconsole] :

    Failover Server Host :

    Agent-Identity Server Shared Secret:

     Re-enter Shared Secret :
```

```
CDSSO Enabled [false]:
-----------------------------------------------------
Agent Configuration file created ==>  defaultConfig
Execute the below command for Agent Configuration :
      cscript.exe IIS6admin.vbs -config defaultConfig
-----------------------------------------------------
```

3. When prompted, provide the following information about the web server instance that this agent will protect:
   - Host Name
   - Web Site Identifier
   - Server Protocol
   - Server Port
   - Agent Deployment URI

   For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.
4. When prompted, provide the following information about the web server that runs Sun ONE Identity Server Services:
   - Primary Server Host
   - Primary Server Protocol
   - Primary Server Port
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host
   - Agent-Identity Server Shared Secret
   - Re-enter Shared secret

- CDSSO feature enabled

For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.

With the information you provide, the script creates the agent configuration file which you can use to configure the agent. For steps to configure the agent, see "Configuring the Agent for Microsoft IIS 6.0 for a Web Site" on page 81.

Creating the Apache 2.0.50 Agent Configuration File

The agent for Apache 2.0.50 provides a Visual Basic (VB) script to help you create the agent configuration file. When you run it, the VB script prompts for information related to, the agent you are installing and Sun ONE Identity Server and creates the agent configuration file based on the information.

NOTE When you are deploying the agent on multiple web sites, you must create a unique agent configuration file for each of the web sites. You can use the following steps to create multiple agent configuration files. Only, make sure that you give a unique file name to each of the configuration files.

To create the agent configuration file, perform the following steps.

1. Change to the directory *Agent_Install_Dir*\apache\bin. This directory stores the VB script required to create the agent configuration file.
2. Run the following command:

   `cscript.exe ApacheCreateConfig.vbs` *defaultConfig*

   where, `ApacheCreateConfig.vbs` is the VB script and *defaultConfig* is the agent configuration file created by this command.

   The script prompts for information as it progresses with the creation of the agent configuration file. The script prompts for information as it progresses with the creation of the agent configuration file. The following is output from running the command.

```
Microsoft (R) Windows Script Host Version 5.6
    Copyright (C) Microsoft Corporation 1996-2001. All rights reserved

    Copyright c 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms
```

```
-----------------------------------------------------
 Apache 2.0.50 Server
-----------------------------------------------------
 Enter the Agent Resource File Name [ApacheResource.en] :

 Fully Qualified Host Name :
 drake.red.iplanet.com

Apache Binary Directory
C:\Program Files\Apache2\bin

Web Server Protocol [http] :
Web Server Port [80] :
Agent Deployment URI [/amagent] :

-------------------------------------------------
Sun Java (TM) Enterprise System Identity Server
-------------------------------------------------
Primary Server Host :
tao.red.iplanet.com

Primary Server Protocol [http]

Primary Server Port Number [80] :

Primary Server Deployment URI [/amserver] :

 Primary Server Console URI [/amconsole] :

 Agent-Identity Server Shared Secret:

 Re-enter Shared Secret :

 CDSSO Enabled [false]:
 -----------------------------------------------------
 Agent Configuration file created ==>  defaultConfig
 Execute the below command for Agent Configuration :
  cscript.exe ApacheAdmin.vbs -config defaultConfig
-----------------------------------------------------
```

3. When prompted, provide the following information about the web server instance that this agent will protect:

   - Host Name
   - Server Protocol
   - Server Port
   - Agent Deployment URI

   For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.

4. When prompted, provide the following information about the web server that runs Sun ONE Identity Server Services:

   - Primary Server Host
   - Primary Server Protocol
   - Primary Server Port
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host
   - Agent-Identity Server Shared Secret
   - Re-enter Shared secret
   - CDSSO feature enabled

   For details on these items, see the corresponding information as described in "Installing and Configuring the Installation Type I Agents" on page 67.

   With the information you provide, the script creates the agent configuration file, which you can use to configure the agent. For steps to configure the agent, see "Configuring the Agent for Apache 2.0.50 for a Web Site" on page 83.

Configuring the Agent for Microsoft IIS 6.0 for a Web Site

Configure the agent for Microsoft IIS 6.0 for a web site after you have created an agent configuration file. If you have not already created an agent configuration file, create one as explained in "Creating the Microsoft IIS 6.0 Agent Configuration File" on page 76.

NOTE If you want to configure the agent for multiple web sites, you must create a separate agent configuration file for each of the web sites.

To configure the agent for a web site, follow these steps:

1. Change to the directory `\Agent_Install_dir\iis6\bin`

2. Run the following command:

   `cscript.exe IIS6admin.vbs -config` *defaultConfig*

   where *defaultConfig* is the agent configuration file.

   The script displays messages to indicate the progress of the configuration as shown in the following sample.

```
Microsoft (R) Windows Script Host Version 5.6
    Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

    Copyright c 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms

    Enter the Agent Resource File Name [IIS6Resource.en] :

    Creating the Agent Config Directory
    Creating the AMAgent.properties File
    Updating the Windows Product Registry
    Loading the IIS 6.0 Agent
    Completed Configuring the IIS 6.0 Agent
```

3. Once the configuration is complete, change to the directory `\`*Agent_Install_Dir*`\iis6\config\`*Identifier_1*

   where *Identifier_1* is the identifier of the web site for which the agent is being configured.

4. Optionally, open the `AMAgent.properties` file and change the value of the property `com.sun.am.logLevels` to `all:5`.

   Before you modify any of the agent properties, refer the appendix Appendix A, "AMAgent Properties" on page 161 for more information.

5. Save the `AMAgent.properties` file.

6. Restart the application pool and the web site.

7. Try accessing the web site (`http://`*fqdn:port*`/index.html`). This link should take you to the Sun ONE Identity Server login page. After a successful authentication, if the policy is properly defined, the user should be able to view the resource.

If you want to view the agent log file `amAgent`, do so at the following location:

`\Agent_Install_Dir\debug\Identifier_`*1*

where Identifier_*1* is the identifier of the web site for which the agent is being configured.

NOTE If you want to configure the agent for multiple web sites, you must follow the above steps for each of the web sites.

Configuring the Agent for Apache 2.0.50 for a Web Site

Configure the agent for Apache 2.0.50 for a web site after you have created an agent configuration file. If you have not already created an agent configuration file, create one as explained in "Creating the Apache 2.0.50 Agent Configuration File" on page 79.

NOTE If you want to configure the agent for multiple web sites, you must create a separate agent configuration file for each of the web sites.

To configure the agent for a web site, follow these steps:

1. Change to the directory `\Agent_Install_dir\apache\bin`
2. Run the following command:

   `cscript.exe Apacheadmin.vbs -config` *defaultConfig*

   where *defaultConfig* is the agent configuration file.

   The script displays messages to indicate the progress of the configuration as shown in the following sample.

```
Microsoft (R) Windows Script Host Version 5.6
    Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

    Copyright c 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms

    Enter the Agent Resource File Name [ApacheResource.en] :

    Creating the AMAgent.properties File
    Modifying httpd.conf
    Completed Configuring the Agent for Apache 2.0.50.
    Re-start your server instance.
```

3. Once the configuration is complete, change to the directory `\Agent_Install_Dir\Apache\config\apache_80`
4. Optionally, open the `AMAgent.properties` file and change the value of the property `com.sun.am.logLevels` to `all:5`.

   Before you modify any of the agent properties, refer to Appendix A, "AMAgent Properties" on page 161 for more information.
5. Save the `AMAgent.properties` file.
6. Change to the directory where the Apache server was installed.
7. Restart the Apache 2.0.50 server.
8. Try accessing the web site (http://drake.red.iplanet.com). This link should take you to the Identity Server login page. After a successful authentication, if the policy is properly defined, the user should be able to view the resource.

   If you want to view the agent log file `amAgent`, do so at the following location:

   `\`*Agent_Install_Dir*`\debug\apache_`*portnumber*

   where *portnumber* is the port number, such as 80, to which the agent is configured.

Using Secure Sockets Layer (SSL) with an Agent

NOTE If your web server is running in SSL and notification is enabled, make sure that you perform the following:

1. Add the server certificate's root CA certificate to the Identity Server's certificate database.
2. Mark the CA root certificate as trusted to enable Identity Server to send notifications to the agent successfully.

For more information on installing a trusted root CA certificate, refer to the documentation for your web server.

During installation, if you choose the HTTPS protocol, the agent is automatically configured and ready to communicate over SSL.

NOTE You should have a solid understanding of SSL concepts and the security certificates required to enable communication over the HTTPS protocol. See the documentation for Sun ONE Web Server at the following location on the Internet:

`http://docs.sun.com/source/816-5682-10/esecurty.htm#1011961`

The Agent's Default Trust Behavior

This section only applies when Identity Server itself is running SSL. By default, the policy agent installed on a supported web server will trust any server certificate presented over SSL by the web server that runs Sun ONE Identity Server; the agent does not check the root Certificate Authority (CA) certificate. If the web server that runs Sun ONE Identity Server is SSL-enabled, and you want the policy agent to perform certificate-checking, you must do two things:

1. Disable the agent's default trust behavior.
2. Install a root CA certificate on the remote web server where the agent is installed. The root CA certificate must the be same one that is installed on the web server that runs Sun ONE Identity Server.

Disabling the Agent's Default Trust Behavior

The following property exists in the `AMAgent.properties` file, and by default it is set to true:

```
com.sun.am.trustServerCerts=true
```

This means that the agent does not perform certificate checking.

To Disable the Default Behavior

The following property must be set to `false`:

```
com.sun.am.trustServerCerts=false
```

Installing the Identity Server Root CA Certificate on the Agent Web Server

The root CA certificate that you install on the web server the agent protects must be the same one that is installed on the web server that runs Sun ONE Identity Server.

Installing the Root CA Certificate on Sun ONE Web Server

See the instructions for installing a root CA Certificate in the documentation that comes with the web server. Generally, this is done through the web server's Administration console. Access the documentation for Sun ONE Web Server 6.0 on the Internet at the following URL:
http://docs.sun.com/source/816-5682-10/esecurty.htm#1011961

Installing the Root CA Certificate on Domino Web Server

The CA certificate that you install on the Domino Web server must be the same one that is installed on the web server that runs Identity Server services.

1. Go to the following directory:

   Agent_Install_Dir\Agents\domino\utils

2. Add the same root CA certificate that is installed on the web server that runs Identity Server services into the existing certificate database. At the command line, enter the following command:

   `certutil -A -n` *cert-name* `-t "C,C,C" -d` *cert-dir* `-i cert-file`

   where:

   - *cert-name* can be any name for this certificate.
   - *cert-dir* is the directory where the certificate and key stores are located. The location is:

   Agent_Install_Dir\Agents\domino\cert

   - *cert-file* is the base-64 encoded certificate file.

   For more information on the `certutil` utility, see the online help by entering the following command:

   `certutil -H`

3. To verify that the root CA certificate was installed properly in the certificate database, enter the following command:

Agent_Install_Dir\bin\certutil -L -d *cert-dir*

You should see the root CA certificate added and listed in the output of the command.

Table 3-3 Example of certutil -L Output

Certificate Name	Trust Attrubutes
cert-name	C,C,C

```
p    Valid peer
P    Trusted peer (implies c)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

4. Restart Domino Web Server.

Installing the Root CA Certificate on Microsoft IIS 5.0

1. Go to the following directory:

   Agent_Install_Dir\iis\cert

2. Add the same root CA certificate that is installed on the web server that runs Sun ONE Identity Server into the existing certificate database. At the command line, enter the following command:

   Agent_Install_Dir\bin\certutil -A -n *cert-name* -t "C,C,C" -d *cert-dir* -i *cert-file*

   where:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located. The location is:

   Agent_Install_Dir\iis\cert

- *cert-file* is the base-64 encoded root CA certificate file.

For more information on the `certutil` utility, see the online help by entering the following command:

```
certutil -H
```

3. To verify that the root CA certificate was installed properly in the certificate database, enter the following command:

   Agent_Install_Dir`\bin\certutil -L -d` *cert-dir*

   You should see the root CA certificate added and listed in the output of the command. See Table 3-3 on page 87 for an example of output after running the `certutil -L` command.

4. Restart IIS.

Installing the Root CA Certificate on Microsoft IIS 6.0

You can use the `certutil` program to install the root CA certificate on Microsoft IIS 6.0.

1. Check if the certificate database is created or not. To do this, open the Microsoft Windows command line and change to the following directory:

   Agent_Install_Dir`\iis\cert`

2. Create the certificate database if you have not already done so, using the following command:

   Agent_Install_Dir`\bin\certutil -N -d .`

3. Install the root CA certificate.

   Agent_Install_Dir`\bin\certutil -A -n` *cert-name* `-t "C,C,C" -d` *cert-dir* `-i` *cert-file*

   where:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located. The location is:

   Agent_Install_Dir`\iis6\cert`

   - *cert-file* is the base-64 encoded root CA certificate file.

   For more information on the `certutil` utility, see the online help by entering the following command:

   ```
   certutil -H
   ```

4. To verify that the certificate is properly installed, at the command line, enter the following:

   Agent_Install_Dir\bin\certutil -L -d *cert-dir*

   You should see the root CA certificate added and listed in the output of the command. See Table 3-3 on page 87 for an example of output after running the `certutil -L` command.

5. Restart IIS.

Installing the Root CA Certificate on Apache 2.0.50

1. Go to the following directory: What is the directory for Apache 2.0.50? I've just started it below?

   Agent_Install_Dir\apache\cert

2. Add the same root CA certificate that is installed on the web server that runs Sun ONE Identity Server into the existing certificate database. At the command line, enter the following command:

   Agent_Install_Dir\bin\certutil -A -n *cert-name* -t "C,C,C" -d *cert-dir* -i *cert-file*

   where:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located. The location is:
     Agent_Install_Dir\apache\cert
   - *cert-file* is the base-64 encoded root CA certificate file.

   For more information on the `certutil` utility, see the online help by entering the following command:

   ```
   certutil -H
   ```

3. To verify that the root CA certificate was installed properly in the certificate database, enter the following command:

   Agent_Install_Dir\bin\certutil -L -d *cert-dir* -i *cert-file*

   You should see the root CA certificate added and listed in the output of the command. See Table 3-3 on page 87 for an example of output after running the `certutil -L` command.

4. Restart Apache 2.0.50 Server

Setting the REMOTE_USER Server Variable

The REMOTE_USER server environment variable is normally set by the agent to the user ID of the user who is accessing the page after being authenticated with Identity Server. By setting this variable to a specific user, the user becomes available to web applications (such as a CGI, servlet, or an ASP program). This feature makes it possible to personalize the content of displayed HTML pages to specific users.

However if the page a user is accessing is not enforced, the REMOTE_USER variable will not be set. To enable setting the REMOTE_USER for not-enforced URLs, you must set

the following property in AMAgent.properties to true (by default the value is false):

```
com.sun.am.policy.agents.anonRemoteUserEnabled=TRUE
```

When you set this property value to TRUE, the value of REMOTE_USER will be set to the value contained in the following property in the AMAgent.properties file (by default, this value is set to anonymous):

```
com.sun.am.policy.agents.unauthenticatedUser=anonymous
```

To enable the REMOTE_USER feature for an IIS 5.0 agent, perform the following steps:

1. From the Windows Start menu, select Programs > Administrative Tools > Internet Services Manager.

   This will launch the Internet Information Services console.

2. On the web site that you want the Sun ONE Identity Server agent to protect, select Properties.

3. Select the Directory Security tab.

4. In the Anonymous Access and Authentication Control section, click Edit.

5. In the dialog that displays, select Anonymous Access and Basic Authentication, then deselect Integrated Windows Authentication.

Validating Client IP Addresses

This feature can be used to enhance security by preventing the stealing or *hijacking* of SSO tokens.

The `AMAgent.properties` file contains a property titled `com.sun.am.policy.agents.client_ip_validation_enable`, which by default, is set to `false`.

If you set this property value to `true`, client IP address validation will be enabled for each in-coming request that contains an SSO token. If the IP address from which request was generated does not match the IP address issued for the SSO token, the request will be denied. This is essentially the same as enforcing a deny policy.

This feature should not be used, however, if the client browser uses a web proxy or if there is a load-balancing application somewhere between the client browser and the agent-protected web server. In such cases, the IP address appearing in the request will not reflect the real IP address on which the client browser runs.

POST Data Preservation

POST data preservation is supported on the Sun ONE Web Server 6.0 SP*x* agent and the Sun ONE Web Server 6.1 agent. Users can preserve POST data, which are submitted to web servers through html forms before users login to the Identity server. Presumably the html page containing the form should be in the not-enforced list. By default, this feature is turned off.

This feature is configurable through two properties in `AMAgent.properties` file. To turn off this feature, use the following `AMAgent.properties` file property and change the value of the property from true to false:

```
com.sun.am.policy.agents.is_postdatapreserve_enabled=true
```

The second property decides how long any POST data can stay valid in the web server cache. After the specified interval, a reaper thread will wake up and clean up any POST cache entries that have lived beyond the specified life time. The following property helps the administrator to configure this time interval. By default this property is set to 10 minutes.

```
com.sun.am.policy.agents.postcacheentrylifetime=10
```

NOTE This feature is not available on the other agents.

Shared Secret Encryption Utility

The policy agent stores the shared secret in the `AMAgent.properties` file. By default, this shared secret is the Identity Server internal LDAP authentication user password. This can be changed on the server side by editing the `AMConfig.Properties` file.

The property `com.sun.am.policy.am.password` in the `AMAgent.properties` file is set with the encrypted shared secret while installing the agent.

To reset or change the shared secret, you can use the encryption utility and set the value in the property.

To reset the shared secret

1. Go to the following directory:

   Agent_Install_Dir`\bin`

2. Execute the following script from the command line

   `cryptit` *shared_secret*

3. Cut and paste the output from Step 2 in the property:

   `com.sun.am.policy.am.password`

4. Restart the web server and try accessing any resource protected by the agent.

Disabling, Uninstalling, and Unconfiguring Microsoft Windows Policy Agents

When you no longer require a Microsoft Windows policy agent, you can disable it, or you can uninstall it. If you no longer require a policy agent to protect a particular web site, you can unconfigure the agent from that web site.

Disabling Microsoft Windows Policy Agents

Microsoft IIS web servers provide a tool that allows you to disable the web server. Therefore, instructions are provided here for disabling the Microsoft IIS web servers using that tool.

Disabling the Policy Agent Installed on Microsoft IIS 5.0

The following steps help you disable an agent installed on Microsoft IIS.

To disable an agent on Microsoft IIS

1. Launch Internet Services Manager.
   - From the Start menu, choose Programs > Administrative Tools > Internet Services Manager.
2. Check the filter status.
   a. Open properties for the host computer in the tree pane of the Internet Services Manager window which is titled “Internet Information Services.”
   b. The host computer name should appear in the tree underneath the Internet Information Services root.
   c. Click Edit in the Master Properties section of the Internet Information Services tab.
   d. Select the ISAPI Filters tab in the WWW Service Master Properties dialog that appears.
   e. Highlight the filter named Sun ONE Identity Server Agent.

      You can click Edit to view the filter name and executable path. You’ll need this information when you want to re-enable the agent. Click Cancel to return to the program.
   f. Click Remove.
   g. Click Apply and exit from the WWW Service Master Properties dialog.
   h. Restart Microsoft IIS.

Disabling the Policy Agent on Microsoft IIS 6.0

Follow these steps to unconfigure the policy agent installed on Microsoft IIS 6.0:

1. From the Microsoft Windows Start menu, choose Programs > Administrative Tools > Internet Information Services Manager.
2. Right click on the web site protected by the policy agent.
3. Open the Properties tab.
4. Click on Home Directory.
5. Click on Configuration.

6. Click *Agent_Install_dir*\iis6\bin\amiis6.dll
7. Click on Remove.
8. Click Yes at the popup “Remove the selected Script Mapping(s)?.”
9. Click Ok.
10. Restart the application pool to which the web site belongs.
11. Restart the web site.

Uninstalling Installation Type I Policy Agents

This section is divided into two subsections explaining how to uninstall Installation Type I policy agents: “Uninstalling Most Installation Type I Policy Agents” on page 94 provides instructions for uninstalling all the Installation Type I policy agents except for Lotus Domino 5 or 6. For these Domino agents, see “Uninstalling the Agents for Lotus Domino 5 or 6” on page 94.

For a list and explanation of the Installation Type I and Installation Type II agents, see “Overview of Policy Agents for Microsoft Windows” on page 64

Uninstalling Most Installation Type I Policy Agents

To uninstall Installation Type I policy agents

1. From the Windows Start menu, choose Settings > Control Panel.
2. In the Control Panel, open Add/Remove Programs.
3. In the Add/Remove Programs window, choose Sun ONE Identity Server Policy Agent.
4. Click Change/Remove.
5. Click Next on Welcome panel.
6. Click Uninstall Now.
7. Click Exit after uninstallation is complete.

Uninstalling the Agents for Lotus Domino 5 or 6

To uninstall the policy agent for Lotus Domino 5 or 6, you should perform the following steps on the Lotus Domino Administrator client from a Windows machine.

To uninstall the agent for Lotus Domino

1. Launch Lotus Domino Administrator.
2. Choose Administrator Tab > Server > All Server Documents.
3. From the listed servers, select the server you want to uninstall.
4. Click Internet Protocols > HTTP tab.
5. Remove the DSAPI filter file name specified for the agent and leave this field blank.
6. Click the Save and Close button to save the changes.
7. Open Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   ```

   ```
   load http
   ```

8. From the Start Menu, choose Settings > Control Panel.
9. In the Control Panel, double-click Add / Remove Programs.
10. In the Add/Remove Programs window, choose Sun ONE Identity Server Policy Agent and click on Change/Remove.
11. In the Welcome Panel, click Next.
12. In the Ready to Uninstall Panel, click Uninstall Now.
13. Click Exit after uninstallation is complete.

Uninstalling Any Agent from the Command-Line

In addition to the GUI mode of uninstallation, the uninstallation program also offers a command-line interface for uninstalling an agent. The command-line interface applies to all agents, Installation Type I and Installation Type II agents. However, Installation Type II agents, first require you to unconfigure the agent from each web site as explained in "Unconfiguring and Uninstalling Installation Type II Policy Agents" on page 96.

To uninstall an agent from the command line

1. In the *Agent_Install_Dir* directory, at the command line, enter the following command:

   ```
   java uninstall_Sun_ONE_Identity_Server_Policy_Agent -nodisplay
   ```

   The uninstallation program displays the following text:

```
1. Uninstall Now
2. Start Over
3. Exit Uninstallation
What would you like to do?
```

2. Enter 1 to start the uninstallation.

   The uninstallation program displays the following text:

```
Product                           Result     More Information
1.  Sun ONE Identity Server Agent  Full       Available
2.  Done
```

3. To see log information, enter 1. To exit the uninstallation program, enter 2.
4. When the uninstallation is completed, you must reboot the system.

   If you want to see more details of the uninstallation, a log file is available in the following location:

   ```
   %TEMP%\Sun_ONE_Identity_Server_Policy_Agent_uninstall*
   ```

Unconfiguring and Uninstalling Installation Type II Policy Agents

If you no longer require an Installation Type II policy agent, Microsoft IIS 6.0 or Apache 2.0.50, to protect a particular web site, you can unconfigure the agent from that web site. Furthermore, if you want to uninstall an Installation Type II agent, you must first unconfigure the agent from all the web sites for which it was configured.

Unconfiguring the Agent for Microsoft IIS 6.0 From a Web Site

Perform the following steps to unconfigure the agent for Microsoft IIS 6.0 from a web site. Make sure that you use the agent configuration file specific to the web site you want to unconfigure.

If you need to unconfigure the agent from multiple web sites, you must repeat these steps for each of the web sites. If you want to uninstall the agent for Microsoft IIS 6.0, you must first unconfigure the agent from all the web sites for which it is configured

1. Stop the web site for which you have configured the agent and the application pool to which the web site belongs.
2. Change to the directory *Agent_Install_Dir*\iis6\bin
3. Run the following VB script to unconfigure the agent:

   `cscript.exe IIS6admin.vbs -unconfig` *defaultConfig*

   where, `IIS6admin.vbs` is the configuration script and *defaultConfig* is the agent configuration file.

   The script unconfigures the agent and displays the following messages.

```
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

Copyright c 2004 Sun Microsystems, Inc. All rights reserved
Use is subject to license terms

Enter the Agent Resource File Name [IIS6Resource.en] :

Removing the Agent Config Directory
Removing the entries from Windows Product Registry
Unloading the IIS 6.0 Agent
Completed Unconfiguring the IIS 6.0 Agent
```

The unconfiguration does the following:

- Removes the agent configuration directory (specific to a web site)
- Removes the entries from Windows registry.
- Removes the wild card application mappings in Microsoft IIS 6.0.

4. Restart the application pool and the web site.

Unconfiguring the Agent for Apache 2.0.50 From a Web Site

Perform the following steps to unconfigure the agent for Apache 2.0.50 from a web site. Make sure that you use the agent configuration file specific to the web site you want to unconfigure.

If you need to unconfigure the agent from multiple web sites, you must repeat these steps for each of the web sites. If you want to uninstall the agent for Apache 2.0.50, you must first unconfigure the agent from all the web sites for which it is configured

1. Stop the web site for which you have configured the agent and the application pool to which the web site belongs.
2. Change to the directory *Agent_Install_Dir*\apache\bin
3. Run the following VB script to unconfigure the agent:

   `cscript.exe Apacheadmin.vbs -unconfig` *defaultConfig*

   where, `Apacheadmin.vbs` is the configuration script and *defaultConfig* is the agent configuration file.

   The script unconfigures the agent and displays the following messages.

```
    Microsoft (R) Windows Script Host Version 5.6
    Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

    Copyright c 2004 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms

    Enter the Agent Resource File Name [ApacheResource.en] :

    Removing the agent configuration directory
    Restoring the original httpd.conf
    Completed Unconfiguring the Agent for Apache 2.050. Re-start your server
instance
```

The unconfiguration does the following:

- Removes the agent configuration directory (specific to a web site)
- Removes the entries from Windows registry.
- Removes the wild card application mappings in Apache 2.0.50.

4. Change to the directory where the Apache server was installed.
5. Restart Apache 2.0.50 server.

Uninstalling Installation Type II Policy Agents

The steps for uninstalling the agent are the same for both Microsoft IIS 6.0 and Apache 2.0.50.

Before running the uninstallation program, ensure that you have already unconfigured the agents from all the web sites for which they were configured as explained in the applicable section:

- “Unconfiguring the Agent for Microsoft IIS 6.0 From a Web Site” on page 97
- “Unconfiguring the Agent for Apache 2.0.50 From a Web Site” on page 98

Please perform the following steps to uninstall the agent:

1. Change to the directory *Agents_Install_Dir*\
2. Run the following command to uninstall the agent:

   ```
   java uninstall_Sun_Java_tm__System_Identity_Server_Policy_Agent
   ```

3. Click Next on the Welcome panel.
4. Click Uninstall Now.

   The program uninstalls the agent.

5. Reboot the computer.

Troubleshooting

Cannot install the agent after a previous installation is removed

The following is an example message displayed when you run the agent installation program:

```
"Sun ONE Identity Server Policy Agent 2.1 for Sun ONE Web Server 6.0 SPx is
installed.  Please refer to installation manual to configure this agent for
another web server instance. Or uninstall it before installing another
agent."
```

Possible Causes:

- You might have an existing installation of the agent.
- You might have a previously-installed agent and did not use the agent's uninstallation program to uninstall the agent.
- The installation program's `productregistry` file may be corrupted.

Solution:

- Check that you have uninstalled any existing installation of the agent.
- The `productregistry` file may be corrupted if there is no existing installation of the agent. This file is used by the installation program to track installed products. It is found in `\WINNT\system32` directory.

NOTE Make a backup copy of this file before you make changes.

- Remove the agent product entry in this file. This entry starts with the following lines:

```
<compid>SUNWamcom
   <compversion>2.1
      <uniquename>SUNWamcom</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>Agent uninstall script
   <compversion>2.1
      <uniquename>Agent uninstall script</uniquename>
      <vendor>Sun Microsystems, Inc.</vendor>
      ......
</compid>
<compid>Agent installer resource bundle
   <compversion>2.1
      <uniquename>Agent installer resource bundle</uniquename>
      <vendor>Sun Microsystems, Inc.</vendor>
      ......
</compid>
<compid>Agent Common Core and SDK
   <compversion>2.1
      <uniquename>Agent Common Core and SDK</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>SUNWames6
   <compversion>2.1
      <uniquename>SUNWames6</uniquename>
      <vendor></vendor>
      ......
</compid>
```

```
<compid>SUNWamcom
<compid>Agent for ...
   <compversion>2.1
      <uniquename>Agent for ...</uniquename>
      <vendor></vendor>
      ......
</compid>
<compid>Sun ONE Identity Server Policy Agent
   <compversion>2.1
      <uniquename>Sun ONE Identity Server Policy Agent</uniquename>
</compid>
```

Unable to uninstall the agent from Windows Start menu > Settings > Control Panel > Add/Remove Programs.

Possible Cause: Java's classpath may not be set correctly on the machine.

Solution: Use the following steps to uninstall the agent.

1. Open Command Prompt Window.
2. Go to *Agent_Install_Dir*
3. Execute command:

   ```
   java uninstall_Sun_ONE_Identity_Server_Policy_Agent
   ```

Lotus Domino 6

Domino Web Server starts with an error message “Unable to load filter”.

Ensure that you have set the Domino DSAPI filter correctly. For steps to do this, see the section “Configuring the Domino DSAPI Filter” on page 71.

Domino DSAPI Filter is not functioning properly on the partitioned server.

Possible Cause: The database you have selected while configuring the DSAPI Filter may be wrong.

Solution: Ensure that you have selected the correct database while configuring the DSAPI filter.

Possible Cause: The partitioned database might not have been updated.

Solution: You might have to replicate the database from the Domino Admin Server.

The agent goes into an infinite loop if the users who are redirected to the resource mentioned in the property com.sun.am.policy.agents.accessDeniedURL do not have the "Get/allow policy" assigned to them.

Solution: Assign "Get/allow policy" to all users of the resource mentioned in the `com.sun.am.policy.agents.accessDeniedURL` property.

Microsoft IIS 5.0 Policy Agent

If you are experiencing problems with your installation, try the following:

- Check the installation log file for errors:

  `%TEMP%\Sun_ONE_Identity_Server_Policy_Agent_uninstall.`*`nnnn`*

- Re-install the agent by uninstalling and then installing.
- Verify agent loading in IIS:

  a. Launch Internet Services Manager.

  b. From the Start menu, choose Programs > Administrative Tools > Internet Services Manager.

  c. Open the properties for the host computer in the Tree Pane of the Internet Services Manager window that is titled Internet Information Services.

  d. The host computer name should appear in the tree underneath the Internet Information Services root.

  e. Click Edit in the Master Properties section of the Internet Information Services tab.

  f. Select the ISAPI Filters tab in the WWW Service Master Properties dialog that appears.

  g. Look for the filter name "Sun ONE Identity Server agent."

  If the Filter name "Sun ONE Identity Server Agent" does not appear at all, then check that the installation program was run, and look for any errors during installation. The install log is located at:

  `%TEMP%\Sun_ONE_Identity_Server_Policy_Agent_uninstall.nnnn`

  A green arrow pointing up in the Status column to the right of the "Sun ONE Identity Server Agent" indicates the agent loaded successfully into IIS. A red arrow pointing down indicates that the filter failed to load. The most likely cause of the filter not loading successfully (red arrow) is that it cannot locate the required dll files.

h. Check your system path to ensure that the following directory is present:

Agent_Install_Dir\bin

i. If the filter did not load successfully check the following:

- Check the path of the Agent DLL by clicking "Sun ONE Identity Server Agent" and then Edit. Ensure that the path in the text box labeled Executable is valid.
- The agent also needs several DLL files. Check that the following exist in the directory Agents\bin:

  amsdk.dll

  ames6.dll

  libnspr4.dll

  libplc4.dll

  libplds4.dll

  libxml2.dll

  nss3.dll

  ssl3.dll

j. If the libraries are in your system path try rebooting the system.

- IIS logs filter loading errors in the System Event Log. To check the event log:

  a. From the Start menu, choose Programs > Administrative Tools > Event Viewer.

  b. Select the System Log.

  c. Check for Error messages with Source W3SVC.

- If the agent loads but returns HTTP 500 Internal Server Error for all URL requests to the IIS web server.

  This indicates that the agent has loaded but did not properly initialize. Returning HTTP 500 Internal Server Error for all HTTP requests is a fail-safe to protect URL resources when the agent cannot initialize. The most likely cause is a Sun ONE Identity Server agent or server misconfiguration or unavailability.

- Check the agent debug log.

The log is located by default at the *Agent_Install_Dir* directory. This is the best source of debug information for resolving initialization and agent operation issues. The log file directory is specified by the property:

`com.sun.am.policy.am.logFile` in the `AMAgent.properties` file located in the directory:

Agent_Install_Dir`\iis\config\`*_PathInstanceName*

The property `com.sun.am.policy.am.loglevels` controls the verbosity of the log information. Set the logging level for the specified logging categories.

The format of the values is:

ModuleName`[:`*Level*`],`*ModuleName*`[:`*Level*`]]*`

The currently used module names are AuthService, NamingService, PolicyService, SessionService, PolicyEngine, ServiceEngine, Notification, PolicyAgent, RemoteLog and all. If the level is omitted, then the logging module will be created with the default logging level, which is the logging level associated with the 'all' module.

The all module can be used to set the logging level for all modules. This will also establish the default level for all subsequently created modules.The meaning of the 'Level' value is described below:

`0` = Disable logging from specified module

`1` = Log error messages

`2` = Log warning and error messages

`3` = Log info, warning, and error messages

`4` = Log debug, info, warning, and error messages

`5` = Like level 4, but with even more debugging messages.

- Check that the agent can locate the `AMAgent.properties` configuration file.

  The agent uses the registry key `HKEY_LOCAL_MACHINE\Software\Sun Microsystems\Identity Server IIS Agent` to locate the `AMAgent.properties` file. The `AMAgent.properties` file is located at:

  Agent_Install_Dir`\iis\config\`*_PathInstanceName*

- The agent uses the Application Event Log to log errors that occur before the debug log file specified in `AMAgent.properties` is started.

  a. From the Start menu, choose Programs > Administrative Tools > Event Viewer.

b. Select the Application Log.

c. Check for agentError messages with source as Sun ONE Identity Server IIS agent.

Known Problems

Agents are not effective after modification of Sun ONE Web Server configuration using the admin console.

The changes made by the agent installation to the server configuration files are overwritten by saving the changes in admin console.

The right procedure when using the admin console should be to load configuration first (from disk file to memory), then make modification, and save the changes (from memory to disk) by clicking the Apply button.

Microsoft IIS 5.0 agent

After installing the policy agent on Microsoft IIS 5.0, stopping individual web sites may occasionally lead to memory corruption messages. You can ignore these messages and restart the Microsoft IIS server.

Chapter 4

Policy Agents on Red Hat, SuSE, and Debian Linux

This chapter explains how to install and configure Web Policy Agents available for the web servers running on Red Hat Linux 7.2, Red Hat Linux 9.0 and Red Hat Advanced Server 2.1.

Topics include:

Before You Begin

Be sure you're familiar with the concepts presented in Chapter 1, "Read This First." The chapter includes brief but important information on the following topics:

Pre-installation Tasks

Before you install the supported agents on the Linux platforms, you must perform the following pre-installation tasks.

Policy Agent for Apache 1.3.27

If you are installing the agent for Apache 1.3.27 on Linux, you must complete the following tasks in the order they are listed below, to make sure Apache Web Server is configured with POSIX Threads library. Failing to perform these tasks may result in the application becoming unusable or may result in the entire system becoming unstable and unusable.

1. Get the Apache source from `http://httpd.apache.org/`

2. Before you run configure, set an environment variable `LIBS=-lpthread` as shown in the table.

Table 4-1 Setting the Environment Variable

Shell	Environment Variable
sh	LIBS=-lpthread;export
bash	export LIBS=-lpthread
tcsh	setenv LIBS '-lpthread'

3. Configure your Apache Web Server with the following flags:

   ```
   configure --enable-rule=SHARED_CORE --enable-shared=max
   ```

4. Rebuild and install Apache Web Server.

5. Install the Apache agent.

Policy Agent for Apache 2.0.48

If you are installing the policy agent for Apache 2.0.48 on Red Hat Advanced Server 2.1, make sure that the following, or a later version, of Red Hat's Program Managers (RPM) are installed on the system for the agent to run correctly:

- `gcc-c++-2.96-124.7.2`
- `glibc-2.2.4-32.11`

The RPMs are available at the following locations:

- `ftp://rpmfind.net/linux/redhat/updates/enterprise/2.1AS/en/os/SRPMS/gcc-2.96-124.7.2.src.rpm`
- `ftp://rpmfind.net/linux/redhat/updates/enterprise/2.1AS/en/os/SRPMS/glibc-2.2.4-32.11.src.rpm`

  This is required because of a bug with exception handling in the C/C++ libraries in the RPM `gcc-2.96-118.7.2` that comes bundled with Red Hat Advanced Server 2.1.

Policy Agents for IBM Lotus Domino 6.0.2 and 6.5

If you are installing the agents for IBM Lotus Domino 6.5 or 6.0.2 on RedHat Advanced Server 2.1, make sure that C++ libraries from GCC 3.3.2 are installed on the system.

This is due to a bug with exception handling in Domino Server on Red Hat Advanced Server 2.1.

The GCC 3.3.2 libraries needed by the agents libraries are:

- `/usr/lib/libstdc++.so.5`
- `/usr/lib/libgcc_s.so.1`

These can be built and installed from GCC 3.3.2 sources available at:

`http://gcc.gnu.org/releases.html`

Policy Agents for Apache 1.3.29 and 2.0.52 on SuSE Linux

The pre-installation information in this section applies specifically to enabling SSL on Apache 1.3.29 and 2.0.52 on SuSE Linux. You first need to download and uncompress the source file.

Obtaining the Apache Server Source File

If you want to compile Apache source by enabling SSL, execute the following command.

```
./configure --prefix=/opt/apache_install --enable-ssl

make

make install
```

Where *apache_install* refers to the directory where the Apache server binary is located.

Stopping and Starting the Apache Server

The command for *starting* the Apache server is different when SSL is enabled. However, the command for *stopping* the Apache server is the same if SSL is enabled or not.

To start the Apache server when SSL is enabled, use the following command.

```
/opt/apache_install/bin/apachetl -D SSL -k start
```

To start the Apache server when SSL is *not* enabled, use the following command.

```
/opt/apache_install/bin/apachetl -k start
```

To stop the Apache server, whether SSL is enabled or not, use the following command.

```
/opt/apache_install/bin/apachetl -k stop
```

Policy Agent for Apache 2.0.52 on Debian Linux

The pre-installation information in this section applies specifically to enabling SSL on Apache 2.0.52 on Debian Linux.

Obtaining OpenSSL

If you want to compile Apache 2.0.52 source for Debian Linux by enabling SSL, you must have OpenSSL on your system. If you do not have SSL on your system, download it from http://www.openssl.org/source/openssl-0.9.7e.tar.gz, open the compressed file, and perform the following steps:

1. In the directory where you downloaded OpenSSL, execute the following command:

```
./config --prefix=/usr/local/openssl

make clean

make install
```

2. Ensure that openSSL is installed at `/usr/local/openssl`.
3. Add `/usr/local/openssl` into your `PATH` variable.

Enabling SSL

To compile Apache source by enabling SSL, follow these steps:

1. Change to the following directory:

   `/opt/apache_src/httpd-2.0.52`
2. Execute the following command:

```
./configure --prefix=/opt/apache_install --enable-ssl

make

make install
```

Where *apache_install* refers to the directory where the Apache server binary is located.

3. Change to the following directory:

   `/opt/`*apache_install*`/conf`
4. Open the `httpd.conf` file.
5. Change the name of the group associated with the `httpd.conf` file from `#-1` to nobody.

Stopping and Starting the Apache Server

The command for *starting* the Apache server is different when SSL is enabled. However, the command for *stopping* the Apache server is the same if SSL is enabled or not.

To start the Apache server when SSL is enabled, use the following command.

```
/opt/apache_install/bin/apachetl -D SSL -k start
```

To start the Apache server when SSL is *not* enabled, use the following command.

```
/opt/apache_install/bin/apachetl -k start
```

To stop the Apache server, whether SSL is enabled or not, use the following command.

```
/opt/apache_install/bin/apachetl -k stop
```

Installing the Agent

The following sections provide step-by-step instructions to help you use the agent installation program in the GUI and the command-line modes.

Installing using the GUI

You must have root permissions when you run the agent installation program.

1. Unpack the product binaries using the following command:

   ```
   # gunzip -dc binaryname.tar.gz | tar -xvof -
   ```

2. Run the `setup` program. You'll find the program in the directory where you untarred the binaries. At the command line, enter the following:

   ```
   # ./setup
   ```

3. In the Welcome page, click Next.
4. Read the License Agreement. Click Yes to agree to the license terms.
5. To search for the directory where you would like to install the agent, click Browse. To accept the default, click Next.

6. When prompted, provide the following information about the web server this agent will protect:

   Install Sun ONE Identity Server Policy Agent in this directory: Enter the full path to the directory where you want this agent to be installed, and then click Next.

   Host Name: Enter the FQDN of the machine where the web server is installed. For example, `mycomputer.example.com`

   Apache Configuration Directory: Specify the Apache server configuration directory where the `httpd.conf` file is located.

   Apache Binary Directory: Enter the full path to the directory where the `httpd` binary is located, for example `usr/local/`*apache2*`/bin`. This field is available only if you are installing the policy agent for Apache 2.0.47 and 2.0.48.

   Web Server Port: Enter the port number for the web server that will be protected by the agent.

   Web Server Protocol: If the web server has been configured for SSL, choose HTTPS; otherwise choose HTTP.

   Agent Deployment URI: Enter a directory name. The default Universal Resource Identifier (URI) is `/amagent`.

   SSL Ready: You should select this option if the Apache web server you are using has support for SSL. If you are using Apache Web Server 1.3.27, your Apache web server is considered SSL ready if it has support for `mod_ssl` and its sources have been compiled using EAPI rule.

   To find out if your Apache web server has been compiled with the EAPI flag, go to the `bin` directory of the Apache web server and type the command:

   ```
   # ./httpd -V
   ```

   You can see various flags that the Apache web server was compiled with. If the flag `-D EAPI` is displayed in this list, it indicates that your Apache Web Server is SSL ready. However, if you do not see this flag; it does not necessarily indicate that the Web Server does not have support for `mod_ssl`.

   The supported configuration for Apache web server are:

   a. Apache Web Server without `mod_ssl` support

   b. Apache Web Server with `mod_ssl` and EAPI flag enabled.

NOTE Apache Web Server with `mod_ssl` support and EAPI flag disabled configuration is not supported by Web Policy Agents.

7. When all the information is entered correctly, click Next.
8. Enter information about the web server that runs Sun ONE Identity Server policy and management. The policy agent will connect to this server.

   Primary Server Host: Enter the FQDN of the system where the primary web server that runs Sun ONE Identity Server is installed. For example, `myserver.example.com`.

   Primary Server Port: Enter the port number for the web server that runs Sun ONE Identity Server.

   Primary Server Protocol: If the web server that runs Sun ONE Identity Server is SSL-enabled, select HTTPS; otherwise select HTTP.

   Primary Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

   Primary Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

   Failover Server Host: Enter the FQDN for the secondary web server that will run Sun ONE Identity Server if the primary web server becomes unavailable. If no failover host exists, then leave this field blank.

   Failover Server Port: Enter the port number of the secondary web server that runs Sun ONE Identity Server. If no failover host exists, then leave this field blank.

   Failover Server Deployment URI: Enter the location that was specified when Sun ONE Identity Server was installed. The default URI for Sun ONE Identity Server is `/amserver`.

   Failover Console Deployment URI: Enter the location that was specified when Sun ONE Identity Server console was installed. The default URI for Sun ONE Identity Server is `/amconsole`.

   Agent Identity Server Shared Secret: Enter the password for the Identity Server internal LDAP authentication user.

   Re-enter Shared secret: Re-enter the password for the Identity Server internal LDAP authentication user.

   CDSSO Enabled: Check this box if you want to enable CDSSO feature.
9. When all the information is entered correctly, click Next.

10. Review the installation summary to be sure that the information you've entered is correct. Note that it displays the CDCServlet URL if you have checked the CDSSO Enabled box in the previous panel. If you want to make changes, click Back. If all the information is correct, click Next
11. In the Ready to Install page, click Install Now.
12. When the installation is complete, you can click Details to view details about the installation, or click Close to close the installation program.
13. Restart your Apache web server for the installation to be complete.

Installing from the Command-Line

You must have root permissions when you run the agent installation program.

1. Unzip the Solaris tar file using the following command:

   ```
   # gunzip -dc binaryname.tar.gz | tar -xvof -
   ```

2. Run the `setup` program. You'll find the program in the directory where you untarred the binaries. At the command line, enter the following:

   ```
   # ./setup -nodisplay
   ```

3. When prompted, provide the following information:

   Have you read, and do you accept, all of the terms of the preceding Software License Agreement? Enter yes.

   Install Sun ONE Identity Server Agent in this directory: Enter the full path to the directory in which you want to install the policy agent.

4. Provide the following information about the Web Server this agent will protect:

   - `Web Server Host Name`
   - `Apache Configuration Directory (Apache Binary Directory if you are installing the agent for Apache 2.0.47 or 2.0.48)`
   - `Web Server Port`
   - `Web Server Protocol`
   - `Agent Deployment URI`
   - `SSL Ready`

   For more information on each of these items, see "Installing using the GUI."

5. Provide the following information about the web server that runs Sun ONE Identity Server:

   - Primary Server Host
   - Primary Server Port
   - Primary Server Protocol
   - Primary Server Deployment URI
   - Primary Console Deployment URI
   - Failover Server Host
   - Failover Server Port
   - Failover Server Protocol
   - Failover Server Deployment URI
   - Failover Console Deployment URI
   - Agent-Identity Server Shared secret
   - Re-enter Shared secret
   - CDSSO Enabled

   For more information on each of these items, see "Installing using the GUI."

   The following text is displayed:

```
Ready to Install

1. Install Now
2. Start Over
3. Exit Installation
```

6. When prompted, **What would you like to do?**, enter 1 to start the installation.

   The following text is displayed:

```
Product                          Result      More Information
1.  Sun ONE Identity Server Agent  Installed   Available
2.  Done
```

7. To see log information, enter 1. To exit the Installation program, enter 2.

Post-installation Tasks

After you have installed the policy agent, you must perform a set of post-installation tasks as explained in the following sections.

Agent for IBM Lotus Domino 6.5

For the agent for IBM Lotus Domino 6.5 to work properly, make sure that the user that Domino server is running as has `read` permission to the following files:

- *Agent_Install_Dir*`/agents/domino6/lib/libamdomino6.so`
- *Agent_Install_Dir*`/domino6/config/`*lotus_notes_notesdata_Dir_separated_by_underscore*`/AMAgent.properties`

  Additionally, the Domino user must have `write` permission to the following file:

- `/var/tmp/debug/`*lotus_notes_notesdata_Dir_separated_by_underscore*`/amAgent`

To set the required permission, you can use the commands as shown in the following examples:

```
chown notes:notes Agent_Install_Dir/SUNWam/agents/domino6/lib/libamdomino6.so
chmown notes:notes
/Agent_Install_Dir/SUNWam/domino6/config/_opt_lotus_notes_notesdata/AMAgent.prop
erties
chown notes:notes  (+w) /var/tmp/debug/_opt_lotus_notes_notesdata/amAgent
```

In the above examples, `notes` is the default user created during IBM Lotus Domino installation.

Additionally, if Sun ONE Identity Server is running with SSL, the files `cert7.db` and `key3.db` must also allow ‘read’ access to the user the Domino Server is running as. These files are available in the directory specified by the property `com.sun.am.sslCertDir` in the `AMAgent.properties` file.

For example, if the property is set as `com.sun.am.sslCertDir = /opt/my-agents-dir`, ensure that `/opt/my-agents-dir/{cert7.db,key3.db}` has the necessary permissions by using the following command:

```
chown notes:notes /opt/my-agents-dir/cert7.db /opt/my-agents-dir/key3.db
```

Configuring the Domino DSAPI Filter

Configure the DSAPI filter as explained here if you are installing the policy agent for IBM Lotus Domino 6.5.1:

1. In Lotus Domino Administrator, choose Administrator Tab > Server > All Server Documents.
2. From the listed servers, select the required server.
3. Click Internet Protocols > HTTP tab.
4. At the DSAPI Filter File Names field, enter *Agent_Install_Dir*`/agents/domino/lib/libamdomino6.so`
5. Click the Save and Close button to save the changes.
6. Open Domino console and restart the server by entering the following commands:

   ```
   tell http quit
   ```

   ```
   load http
   ```

Configuring the Agent for Multiple Web Server Instances

To configure an agent for multiple web server instances on a single computer, use the GUI or the command-line version of the agent installation program to install the first agent. After the first agent is installed, you can then install successive agents using the `config` script. This script must be run from the command line as described in the following section.

Configuring the Agent for Multiple Web Server Instances on the Same Computer

Once you have installed an agent on your system, you can configure it for multiple web server instances on that system using a script that is copied to the system during the agent installation. The two scripts, `config_linux` and `unconfig_linux`, located in the following directory:

Agent_Install_Dir`/agents/apache/bin`

To configure additional agents on a system after the original agent has been installed, run the `config_linux` script from the `bin` directory using the following command:

```
# ./config_linux
```

Follow the prompts to install the additional agents. For information on each of the prompts, see "Installing using the GUI." In general, information needs to be entered for both the protected Apache server instance and the Sun ONE Identity Server server(s). The following text shows an example run:

```
# ./config_linux
Enter the Apache Server Configuration Directory: [/etc/httpd/conf]
SSL Ready: [true] false
Enter the Local Hostname: [mycomputer.example.com]
Enter the Agent Web Server Port: [80]
Select Agent Web Server Protocol: [1] http [2] https-->[1]
Enter the Agent Deployment URI: [/amagent]
Select Identity Server Protocol: [1] http [2] https --> [1]
Enter the Identity Server Hostname: [mycomputer.example.com]
Enter the Identity Server Port: [58080]
Enter the Identity Server Deployment URI [/amserver]
Enter the Identity Server's Console Deployment URI [/amconsole]
Select Failover Identity Server Protocol: [1] http [2] https [3] no failover
--> []
Enter the Failover Identity Server Hostname: []mycomputer.example.com
Enter the Failover Identity Server Port: []
Enter the Identity Server Deployment URI [/amserver]
Enter the Identity Server's Console Deployment URI [/amconsole]
Enter Agent-Identity Server shared secret:
Re-enter Agent-Identity Server shared secret:
Is CDSSO Enabled: [1] yes [2] no --> [2]
Configuring Apache Web Server ...
Done
```

NOTE Be sure to use the `unconfig_linux` script to uninstall any agent that was installed using the `config_linux` script—you cannot use the GUI installation program to uninstall agents that were installed from the command line. The GUI uninstallation program must be executed only after unconfiguring all the existing agents installed using command-line `unconfig` script.

Using Secure Sockets Layer (SSL) with an Agent

During installation, if you had chosen the HTTPS protocol, the agent is automatically configured and ready to communicate over SSL.

NOTE Before proceeding with the following steps, ensure that the Web Server is configured for SSL.

The Agent's Default Trust Behavior

By default, the policy agent installed on a remote Apache Server will trust any server certificate presented over SSL by the Web Server that runs Sun ONE Identity Server; the agent does not check the root Certificate Authority (CA) certificate. If the Web Server that runs Sun ONE Identity Server is SSL-enabled, and you want the policy agent to perform certificate-checking, you must do the following:

1. Disable the agent's default trust behavior.
2. Install a root CA certificate on the remote web server (where the agent is installed). The root CA certificate must the be same one that is installed on the web server that runs Sun ONE Identity Server service.

Disabling the Agent's Default Trust Behavior

The following property in the `AMAgent.properties` file controls the agent's trust behavior. By default it is set to `true`:

```
com.sun.am.trustServerCerts=true
```

This means that the agent does not perform certificate-checking.

To Disable the Default Behavior

The following property must be set to `false`:

```
com.sun.am.trustServerCerts=false
```

Installing the Root CA Certificate on the Remote Web Server

The root CA certificate that you install on the remote web server must be the same as the one installed on the web server that runs Sun ONE Identity Server.

To Install the Root CA Certificate

You can use the `certutil` program to install the root CA Certificate on Apache web server.

1. In C shell, at the command line, enter the following commands (assuming `/etc/httpd/apache` is the directory where the Apache `config` file is located):

   ```
   # cd /etc/apache/cert
   ```

   ```
   # setenv LD_LIBRARY_PATH
   /Agent_Install_Dir/agents/apache/lib:/Agent_Install_Dir/agents/lib:/usr/lib/mps
   ```

2. Create the necessary certificate database if you have not already done.

   ```
   # /Agent_Install_Dir/agents/apache/cert/certutil -N -d .
   ```

3. Install root CA certificate.

   ```
   # /Agent_Install_Dir/agents/apache/cert/certutil -A -n cert-name -t "C,C,C" -d
   cert-dir -i cert-file
   ```

   In the commands above, the variables represent the following:

   - *cert-name* can be any name for this root CA certificate.
   - *cert-dir* is the directory where the certificate and key stores are located.
   - *cert-file* is the base-64 encoded root CA certificate file.

   For more information on the `certutil` utility, enter `certutil -H` for online Help.

4. To verify that the certificate is properly installed, at the command line, enter the following:

   ```
   # ./certutil -L -d .
   ```

   Trust database information will include the name of the root CA certificate you installed. Example:

```
Certificate Name                                Trust Attrubutes

cert-name                                       C,C,C

p    Valid peer
P    Trusted peer (implies c)
c    Valid CA
T    Trusted CA to issue client certs (implies c)
C    Trusted CA to certs(only server certs for ssl) (implies c)
u    User cert
w    Send warning
```

Setting the REMOTE_USER Server Variable

The REMOTE_USER server environment variable can be set to a Sun ONE Identity Server authenticated user or an anonymous user. By setting this variable to a specific user, the user becomes available to web applications (such as a CGI, servlet, or ASP program). This feature makes it possible to personalize the content of displayed HTML pages to specific users.

To enable the REMOTE_USER setting for globally not-enforced URLs as specified in the AMAgent.properties file (these are URLs that can be accessed by non-authenticated users), you must set the following property in the AMAgent.properties file to TRUE (by default, the value of this property is set to FALSE):

```
com.sun.am.policy.agents.anonRemoteUserEnabled=TRUE
```

When you set this property value to TRUE, the value of REMOTE_USER will be set to the value contained in the following property in the AMAgent.properties file (by default, this value is set to anonymous):

```
com.sun.am.policy.agents.unauthenticatedUser=anonymous
```

Validating Client IP Addresses

This feature can be used to enhance security by preventing the stealing or *hijacking* of SSO tokens.

The `AMAgent.properties` file contains a property named `com.sun.am.policy.agents.client_ip_validation_enable`, which by default is set to `false`.

If you set this property to `true`, client IP address validation will be enabled for each in-coming request that contains an SSO token. If the IP address from which request was generated does not match the IP address issued for the SSO token, the request will be denied. This is essentially the same as enforcing a deny policy.

This feature should not be used, however, if the client browser uses a web proxy or if there is a load-balancing application somewhere between the client browser and the agent-protected web server. In such cases, the IP address appearing in the request will not reflect the real IP address on which the client browser runs.

Shared Secret Encryption Utility

The policy agent stores the shared secret in the `AMAgent.properties` file. By default, this password is the Identity Server internal LDAP authentication user password. This can be changed on the server side by editing the `AMConfig.Properties` file.

The property `com.sun.am.policy.am.password` in the `AMConfig.Properties` file is set with the encrypted shared secret while installing the agent.

To reset or change the shared secret, you can use the following utility and set the value in the property.

1. Go to the following directory:

   Agent_Install_Dir`/bin`

2. Execute the following script from the command line:

   `crypt_util` *shared_secret*

3. Cut and paste the output from Step 2 in the property:

   `com.sun.am.policy.am.password`

4. Restart the Web Server and try accessing any resource protected by the agent. If the agent gets redirected to the Sun ONE Identity Server, this indicates the above steps were executed properly.

Uninstalling the Policy Agent

The following sections explain the procedure to uninstall a policy agent.

Removing an Agent using the unconfig Script

To remove an agent that was installed from the command line using the `config_linux` script, use the script `unconfig_linux`. The `unconfig_linux` script is located in the following directory:

Agent_Install_Dir`/agents/apache/bin`

Here is an example run of the `unconfig_linux` script.

```
# ./unconfig_linux /web_server_root/httpd/conf
Unconfiguring webserver ...
done.
```

Uninstalling using the GUI

Use the following steps to uninstall the policy agent using the GUI of the installation program:

1. In the directory where the agent is installed, at the command line, enter the following command:

   ```
   # ./uninstall_linux_apache_agent
   ```

2. Click Next on Welcome panel.
3. Click Uninstall Now.
4. Click Close after uninstallation is complete.
5. Restart the web server.

Uninstalling from the Command Line

Use the following steps to uninstall the policy agent from the command line:

1. In the *Agent_Install_Dir* directory, at the command line, enter the following command at the command line:

```
# ./uninstall_linux_apache_agent -nodisplay
```

The following text is displayed:

```
The uninstaller has detected the following agents on this system:
1. Agent 2.1 for Apache [/usr/local]
2. Exit
Please select an installed agent from the following list:
```

2. Enter `1`, to remove the product.

   The following text is displayed:

```
Ready to Uninstall

1. Uninstall Now
2. Start Over
3. Exit Uninstallation
```

3. When prompted, **What next?** enter `1` to begin uninstallation.

   The following text is displayed:

```
Product                                 Result   More Information
1. Sun ONE Identity Server Policy Agent  Full     Available
2. Done
```

4. To see log information on the agent, enter `1`. To exit the uninstallation program, enter `2`.
5. Restart the web server.

Troubleshooting

Error message displayed during startup

Apache server displays the following error message during startup after the agent is installed:

```
Syntax error on line 1 of

/etc/opt/SUNWam/agents/apache/config/_usr_local_apache_conf/dsame.conf:

Invalid command 'LoadModule', perhaps mis-spelled or defined by a module
not included in the server configuration

./apachectl start: httpd could not be started
```

Solution: This indicates that the Apache server does not have mod_so enabled and consequently does not support dynamic shared objects. To enable mod_so support, refer to Apache server documentation at http://httpd.apache.org/

Chapter 5

Single Sign-on Solution for Oracle Application Servers

This chapter explains how you can deploy the Single Sign-on (SSO) solution for Oracle9iAS R1 and Oracle Application Server 10g using Sun ONE Identity Server. This chapter includes the following topics:

Introduction

Oracle9iAS R1 and Oracle Application Server 10g use Login Server and Oracle Single Sign-On (SSO) Server respectively to provide SSO functionality to its partner applications. When a user presents credentials to a partner application, such as Oracle Portal, the request is redirected to the Login Server / Oracle SSO Server, which upon successful authentication, issues a token that represents the user's identity. This token is passed on to the calling partner application, which then creates its own session tokens. Upon logout, the partner application performs its own local logout procedures and then immediately calls the Login Server / Oracle SSO server's logout procedure. On completion of these procedures, the user's session is invalidated.

Integration with Sun ONE Identity Server

Oracle 9iAS R1 or Oracle Application Server 10g can be integrated with Sun ONE Identity Server to achieve SSO functionality. In this type of integration, Sun ONE Identity Server sits in front of Login Server / Oracle SSO Server and provides user authentication only. Login Server or Oracle SSO Server is still responsible for managing user accounts, checking account policies, auditing, generating tokens, and redirecting users to partner applications. Since Login Server / Oracle SSO Server is still responsible for generating trust tokens, special steps must be taken to ensure that it obtains the user's identity from Sun ONE Identity Server after successful authentication.To accomplish this task, Sun ONE Identity Server must pass the identity of the user to Login Server / Oracle SSO Server via HTTP headers. Once Login Server / Oracle SSO Server retrieves a user's identity, it can generate a trust token for the partner applications and SSO will function normally within the environment. Since Login Server / Oracle SSO Server handles all of these responsibilities, the partner applications, which interact solely with Login Server / Oracle SSO Server, are completely unaware of the integration with Sun ONE Identity Server.

Software Requirements

This SSO solution requires the following software to be installed.

For Oracle9iAS R1

Table 5-1 Software Requirements for the SSO Solution for Oracle9iAS R1

Software	Supported Platforms
Sun ONE Identity Server, version 6.1	Solaris 8 and 9
Web Policy Agent for Oracle9iAS R1 Apache 1.3.29	Solaris 8 and 9
Oracle9iAS R1	Solaris 8 and 9

For Oracle Application Server 10g

Table 5-2 Software Requirements for the SSO Solution for Oracle Application Server 10g

Software	Supported Platforms
Sun ONE Identity Server, version 6.1	Solaris 8 and 9
Web Policy Agent, version 2.1	Solaris 8 and 9
Oracle Application Server 10g	Solaris 8 and 9

Deploying the Integrated SSO Solution

The following sections present steps to deploy the integrated SSO solution for Oracle9iAS R1 and Oracle Application Server 10g. Click one of these links to go to the appropriate section.

- Deploying the Solution for Oracle9iAS R1
- Deploying the Solution for Oracle Application Server 10g

Deploying the Solution for Oracle9iAS R1

Do the following to deploy the integrated Sun ONE Identity Server SSO solution for Oracle9iAS:

1. Set up two computers as follows:
    - On the first computer, install Oracle9iAS R1, which includes Login Server and Oracle Portal Server. For detailed installation steps, refer Oracle documentation.
    - On the second computer, install Sun ONE Identity Server, version 6.1, and Sun ONE Directory Server. Now, synchronize Sun ONE Directory Server with Oracle user store on the first computer. For example, if user *portal30_sso* is a valid user in Oracle, that user should also be a valid user in Sun ONE Directory Server. One of the ways to synchronize user data is by using Oracle's synchronization connector for Sun ONE Directory Server. Refer to the Sun ONE Identity Server and Sun ONE Directory Server documentation for more details on adding users to Sun ONE Identity Server.

2. Modify the default external authentication implementation, `wwsso_auth_external`, in the file `ssoxnete.pkb`. This file can be found in the directory `$ORACLE_HOME/portal30/admin/plsql/sso`. The only two functions that must be modified here are `authenticate_user` and `get_Authentication_Name`. The modified functions are documented below with the changes highlighted in bold.

Code Example 5-1 Changes to the Function authenticate_user

```
FUNCTION authenticate_user
  (
   p_user OUT VARCHAR2
  )
  return PLS_INTEGER
IS
 l_http_header varchar(1000);
 l_ssouser wwsec_person.user_name%type := NULL;
BEGIN

    l_http_header := owa_util.get_cgi_env('HTTP_IDENTITY_USER');
    debug_print('Identity ID : ' || l_http_header);

   l_ssouser := l_http_header;
IF ( (l_ssouser IS NULL) or
       ( INSTR(l_ssouser, GLOBAL_SEPARATOR) != 0) ) THEN
       debug_print('malformed user id: '
               || l_ssouser
               || ' returned by wwsso_auth_external.authenticate_user');
       RAISE EXT_AUTH_FAILURE_EXCEPTION;
   ELSE
     p_user := NLS_UPPER(l_ssouser);
     return 0;
   END IF;
EXCEPTION
   WHEN OTHERS THEN
   WHEN OTHERS THEN
     debug_print('unknown exception in authenticate_user(p_user)'
                 || sqlerrm);
     RAISE EXT_AUTH_FAILURE_EXCEPTION;

END authenticate_user;
```

Code Example 5-2 Changes to the Function get_Authentication_Name

```
FUNCTION get_authentication_name
 RETURN VARCHAR2
AS
```

Code Example 5-2 Changes to the Function get_Authentication_Name

```
BEGIN
    RETURN 'Sun ONE Identity Server';
END get_authentication_name;
```

3. Run `ssonete.sql` on the first computer to configure Login Server to operate in the external mode and load the new external authentication implementation for Sun ONE Identity Server, which was just saved to `ssoxnete.pkb` in the previous step. This file can be found in the directory `$ORACLE_HOME/portal30/admin/plsql/sso` and should be run as user `portal30_sso`.
4. Restart Login Server for the changes to take effect.
5. Install the policy agent for Apache Web Server 1.3.29 in the same machine that Oracle 9iAS R1 is installed. The agent module gets installed on the same instance of Login Server. For instructions to install the agent, see Chapter 2, "Policy Agents on Solaris and HP-UX". When installing the agent, the SSL Ready box must be checked since Oracle's Apache Server supports SSL. Additionally, make sure CDSSO is enabled even if the agent and Sun ONE Identity Server are in the same domain.
6. Configure the agent by modifying the `AMAgent.properties` files. The section Configuring the Agent explains the properties that must be changed.
7. Edit the `modplsql` Database Access Descriptor (DAD) file named `wdbsvr.app` for Oracle9iAS on the first computer, to include the following entries. This file is located in the directory `$ORACLE_HOME/Apache/modplsql/cfg`.

   - Identity Server headers:

     `cgi_env_list = HTTP_IDENTITY_USER`

   - Connect string for the Login Server schema:

     `connect_string` = *FQDN*:*Database_Listener_Port*:*ORACLE_SID*

   The `cgi_env_list` and `connect_string` entries must be added under the sections `[DAD_portal30]` and `[DAD_portal30_sso]`.

Deploying the Solution for Oracle Application Server 10g

Follow these steps to deploy the SSO Solution for Oracle Application Server 10g. For this SSO solution, Oracle Portal was used as a partner application to verify the integration.

1. Install Oracle Application Server 10g including Oracle SSO Server and Oracle Portal.

   Make sure that Oracle Application Server 10g is installed using a fully qualified domain name (FQDN) for the server. If it is not installed using FQDN, third-party authentication will not work. Oracle Application Server 10g can be installed using FQDN by running the installation program with the following option:

   `./runInstaller OUI_HOSTNAME=`*`server.domain`*

   For example, if the hostname of your machine is *agent1* and the domain is *example.com*, then the Oracle installation program should be started with the following command:

   `./runInstaller OUI_HOSTNAME=`*`agent1.example`*`.com`

   Consult Oracle Application Server 10g documentation for further details.

2. Install Sun ONE Identity Server, version 6.1, on a separate server.

3. Create the necessary users in Oracle Internet Directory (OID), with the proper roles, for your environment.

4. Synchronize the Oracle users in OID with users in Sun ONE Directory Server. This can be done by manually adding the OID users to Sun ONE Directory Server or by using the OID Connector for Sun ONE Directory Server, which synchronizes users between the two stores. Basically, if a user name exists in OID, then that user name must also exist in Sun ONE Directory Server. Refer to the Sun ONE Identity Server / Sun ONE Directory Server documentation for more details on adding users manually or to the OID Connector documentation for more information on automatic synchronization between the two user stores. You can download the OID Connector for Sun ONE Directory Server at:

   http://download-west.oracle.com/docs/cd/B10464_02/manage.904/b12118/odip_ipl.htm#122580

5. Install the policy agent, version 2.1 for Apache 1.3.29 on the same machine where Oracle SSO Server is running. The policy agent for Apache 1.3.29 can be downloaded from Sun at:

   http://wwws.sun.com/software/download/allproducts.html#id_server_agents

   Refer Chapter 2, "Policy Agents on Solaris and HP-UX" of this guide for detailed instructions to install the agent. When the installation prompts you for an Apache instance, select the Oracle Single Sign-On server instance, which was installed with the midtier. This allows the agent to protect the Oracle SSO Server. Additionally, the SSL Ready box must be checked since Oracle's Apache is SSL ready.

6. Configure the policy agent for Apache 1.3.29 by modifying the agent's `AMAgent.properties` file. Refer the section Configuring the Agent for more information about the properties you should modify.

7. Once you have set the values for agent properties, create a file called `SSOTPAMAuth.java` in the directory `$<ORACLE_HOME_INFRASTRUCTURE>/j2ee/OC4J_SECURITY/applications/sso/web/WEB-INF/classes`

   Here is a sample file that you can use. You should use the exact contents of this file except the URL returned by the method `getUserCredentialPage`. This method should be configured to return an error URL that is pertinent to your environment. In this example, you should substitute `error.html` with an error URL from your environment. Additionally, you can add extra debug information in this file or change the exception output to help diagnose any problems arising out of third-party authentication.

Code Example 5-3 `SSOTPAMAuth.java`

```
 /**
  * returns IPASUserInfo
  /**

  /* Copyright (c) 2002, 2003, Oracle Corporation. All rights reserved. */
 /*
  DESCRIPTION
          Class for Sun ONE  Identity Server integration with SSO Server

  PRIVATE CLASSES
 NOTES
  This class implements the SSOServerAuthInterface. To enable this
 integration, replace: oracle.security.sso.server.auth.SSOServerAuth with
 oracle.security.sso.server.auth.SSOTPAMAuth for the desired security level
 in policy.properties.
  */
 package SunTPAM.security.ssoplugin;
```

Code Example 5-3 SSOTPAMAuth.java

```
/**
import java.io.PrintWriter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import oracle.security.sso.ias904.toolkit.IPASAuthInterface;
import oracle.security.sso.ias904.toolkit.IPASAuthException;
import oracle.security.sso.ias904.toolkit.IPASUserInfo;
import oracle.security.sso.ias904.toolkit.IPASInsufficientCredException;
import java.net.URL;
public class SSOTPAMAuth implements IPASAuthInterface {

 private static String CLASS_NAME = "SSOTPAMAuth";
 private static String TPAM_USER_HEADER = "identity-user";

 public SSOTPAMAuth() {

 }
public IPASUserInfo authenticate(HttpServletRequest request) throws
IPASAuthException, IPASInsufficientCredException {

String TPAMUserName = null;

 try {

     TPAMUserName = request.getHeader(TPAM_USER_HEADER);
} catch (Exception e) {

                throw new IPASInsufficientCredException("No TPAM Header");
 }

 if (TPAMUserName == null) throw new IPASInsufficientCredException("No TPAM
Header");

 IPASUserInfo authUser = new IPASUserInfo(TPAMUserName);
 return authUser;

 }
public URL getUserCredentialPage(HttpServletRequest request, String msg)
 {
 // This function will never have been reached in the case of TPAM
 // as the TPAM Agent will intercept all requests

try {

   return new URL("<error.html>");
} catch (Exception e) {

System.out.println("Exception in SSOTPAMAuth");
e.printStackTrace();

}
System.out.println("Error encountered in SSOTPAMAuth");
return null;
 }
}
```

8. Add the following lines to your CLASSPATH environment variable:

   $*ORACLE_HOME_INFRASTRUCTURE*/j2ee/home/lib/servlet.jar:$<ORACLE_HOME_INFRASTRUCTURE>/sso/lib/ipastoolkit.jar

   Note that $*ORACLE_HOME_INFRASTRUCTURE* must be replaced to point to the ORACLE_HOME directory where Oracle 10g Infrastructure is installed.

9. Go to the directory $*ORACLE_HOME_INFRASTRUCTURE*/j2ee/OC4J_SECURITY/applications/sso/web/WEB-INF/classes and compile the file SSOTPAMAuth.java with the following command:

   ```
   javac -d . SSOTPAMAuth.java
   ```

   This should create the directory structure SunTPAM/security/ssoplugin and place the compiled SSOTPAMAuth class there.

10. Copy the entire directory structure *SunTPAM/security/ssoplugin* and its contents to the directory $*ORACLE_HOME_INFRASTRUCTURE*/sso/plugin. The file SSOTPAMAuth.class that you just compiled should now be under $*ORACLE_HOME_INFRASTRUCTURE*/sso/plugin/SunTPAM/security/sssoplugin.

11. Modify the file policy.properties located in the following directory:

    $*ORACLE_HOME_INFRASTRUCTURE*/sso/conf

    In this file, change the property MediumSecurity_AuthPlugin so as follows:

    ```
    MediumSecurity_AuthPlugin = SunTPAM.security.ssoplugin.SSOTPAMAuth
    ```

    This property specifies the authentication module for Sun ONE Identity Server that Oracle SSO Server must use.

    No other parameters in this file need to be changed for this integration to be successful. However, if need be, the property debugLevel can be changed to provide more debug information for the Oracle SSO server, if necessary. The Oracle SSO server provides the following four levels of debugging:

    - ERROR: log errors only
    - WARN: log both errors and warning messages
    - INFO: log messages such as current data and time, for instance, as well as errors and warnings

- DEBUG: log details about program execution as well as errors, warnings, and messages

The Oracle SSO Server debug file is defined by the property `debugFile`. The debug file provides debugging information to the Oracle SSO Server. For further information on how this file should be set, please refer to the Oracle Application Server 10g documentation.

Additionally, if you want to see where Java exceptions and `System.out.println` lines from the `SSOTPAMAuth` class are logged, you can see them at the following location: `$ORACLE_HOME/opmn/logs/OC4J~OCJ4_SECURITY~default_island~1`.

Configuring the Agent

You can configure the agent by modifying the following properties in the file `AMAgent.properties`. The `AMAgent.properties` file can be found in the following directory:

`/etc/opt/SUNWam/agents/apache/config/`*`_Pathinstancename`*`/`

The following sections explains the properties you need to set for Oracle9iAS R1 and Oracle Application Server 10g.

Policy Agent for Oracle9iAS R1

To configure the agent for Oracle9iAS R1, you must modify the following properties in the file `AMAgent.properties`:

fetchHeaders

Set this value to `true` so that additional policy response attributes can be introduced into the HTTP headers.

```
com.sun.am.policy.am.fetchHeaders=true
```

headerAttributes

This value represents what policy attributes should be added to the HTTP header (if the value of `fetchHeaders` is `true`). Set this value to `uid|identity-user` so that the user id of Sun ONE Identity Server is passed to Login Server.

```
com.sun.am.policy.am.headerAttributes=uid|identity-user
```

do_sso_only

Set this value to `true` so that the agent will just enforce user authentication (SSO) without enforcing policies (authorization). In this integration, Login Server handles authorization.

```
com.sun.am.policy.agents.do_sso_only = true
```

fqdnDefault

This value is set by the agent installation program to the hostname where the agent is installed. Make sure that this value is a fully qualified domain name. It should be set to *hostname.domain*. For example, if the machine is called *agent1* and the domain is *example*`.com`, this property should be set as follows:

```
com.sun.am.policy.agents.fqdnDefault = agent1.example.com
```

fqdnMap

This value must be set manually after the agent is installed. It should be set to *hostname*`|`*hostname*, where *hostname* is not a fully qualified domain name and represents the machine where the agent is installed. If the machine is called *agent1* and the fully qualified domain name of the system is *agent1.example.com*, the fqdnMap property should be set as follows:

```
com.sun.am.policy.agents.fqdnMap = agent1 | agent1
```

For more information about this property see “com.sun.am.policy.agents.fqdnMap” on page 174)

cdsso-enabled

If CDSSO was enabled when the agent was installed, this property will be set automatically. Otherwise, add this property and enable it so that SSO works properly in the Oracle environment.

```
com.sun.am.policy.agents.cdsso-enabled = true
```

cdcservletURL

If CDSSO was enabled when the agent was installed, this property will be set automatically. Otherwise, add this property and enable it so that SSO works properly in the Oracle environment. It should be set to `http://`*FQDN hostname*`:`*port*`/amserver/cdcservlet` where *FQDN hostname* is the fully qualified

hostname where Sun ONE Identity Server is installed and *port* is the port where the amserver process is running. For example, if Sun ONE Identity Server is installed on the machine *agent2.example.com* and listens on port 58080, then this property should be set as follows:

```
com.sun.am.policy.agents.cdcservletURL =
http://agent2.example.com:58080/amserver/cdcservlet
```

reverse_the_meaning_of_notenforcedList

Set this value to true so that the notenforcedList becomes the enforced list.

```
com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList = true
```

notenforcedList

When the SSO integration is performed with Oracle Portal 3.0.9, this value must be set to the login pages of both Login Server and Oracle Portal 3.0.9 as follows:

```
http://hostname:port/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.login?p_req
uested_url=http%3A%2F%2Fhostname%3Aport%2Fpls%2Fportal30_sso%2FPORTAL30_SS
O.home&p_cancel_url=http%3A%2F%2Fhostname%3Aport%2Fpls%2Fportal30_sso%2FPO
RTAL30_SSO.home
```

```
http://hostname:port/pls/portal30/PORTAL30.wwsec_app_priv.login?p_requested_u
rl=http%3A%2F%2Fhostname%3Aport%2Fpls%2Fportal30%2FPORTAL30.home&p_cancel_
url=http%3A%2F%2Fhostname%3Aport%2Fpls%2Fportal30%2FPORTAL30.home
```

In these examples, *hostname* is the hostname of the system where the agent is installed and *port* is the port where the Oracle HTTP Server is running. For example, if the hostname is *agent1* and the port is *7779*, this value should be set as follows:

```
com.sun.am.policy.agents.notenforcedList =
http://agent1:7779/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.login?p_re
quested_url=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30_sso%2FPORTAL30_SSO
.home&p_cancel_url=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30_sso%2FPORTA
L30_SSO.home
```

```
http://agent1:7779/pls/portal30/PORTAL30.wwsec_app_priv.login?p_requested_
url=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30%2FPORTAL30.home&p_cancel_u
rl=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30%2FPORTAL30.home
```

NOTE If additional Oracle Partner applications are used, then the login pages of those applications must also be added to this list.

logout.url

This value specifies the logout URLs of Login Server and the partner applications. These URLs are never enforced by the agent. When the agent sees any of these URLs, it checks whether a valid session ID for the user still exists. If one does exist, the agent invalidates it and effectively logs the user out of Sun ONE Identity Server. The agent then passes the request onto Login Server so that the logout can be processed there.

In this integration, the logout URL for Login Server and Oracle Portal 3.0.9 are included since Oracle Portal is the partner application chosen for verification. Note that these values are separated by only a space. Here is an example:

```
com.sun.am.policy.agents.logout.url =
http://hostname:port/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.logout?p_done_url=http%3A%2F%2F<hostname>%3A<port>%2Fpls%2Fportal30_sso%2FPORTAL30_SSO.home http://hostname:port/pls/portal30/PORTAL30.wwsec_app_priv.logout?p_done_url=http%3A%2F%2F<hostname>%3A<port>%2Fpls%2Fportal30%2FPORTAL30.home
```

Here, the *hostname* refers to the host where the agent is installed and the *port* is the port of the Oracle HTTP Server. For example, if the hostname is *agent1* and the port is *7779*, then this value should be set as follows:

```
com.sun.am.policy.agents.logout.url=http://agent1:7779/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.logout?p_done_url=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30_sso%2FPORTAL30_SSO.home

http://agent1:7779/pls/portal30/PORTAL30.wwsec_app_priv.logout?p_done_url=http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30%2FPORTAL30.home
```

NOTE If additional Oracle Partner applications are used, then the logout pages for those applications must also be added to this list.

logout.cookie_reset_list

This property lists the cookies that need to be reset or removed upon log out from Login Server. If Oracle Portal 3.0.9 is also used for the integration, cookies for both Login Server and Oracle Portal must be present in this list as follows:

```
com.sun.am.policy.agents.logout.cookie_reset_list = Domain=,
iPlanetDirectoryPro, iPlanetDirectoryPro;Domain=, portal30,portal30;Domain=
```

NOTE If additional Oracle partner applications are used, then cookies for those applications must also be added to this list so they can be reset/removed upon logout from Login Server.

Policy Agent for Oracle Application Server 10g

You can configure the agent for Oracle Application Server 10g by modifying the following properties in the file `AMAgent.properties`:

fetchHeaders

Set this value to `true` so that additional policy response attributes can be introduced into the HTTP headers.

```
com.sun.am.policy.am.fetchHeaders=true
```

headerAttributes

This value represents what policy attributes should be added to the HTTP header (if the value of `fetchHeaders` is `true`). Set this value to `uid|identity-user` so that the user id of Sun ONE Identity Server is passed to Oracle SSO Server via HTTP headers.

```
com.sun.am.policy.am.headerAttributes=uid|identity-user
```

do_sso_only

Set this value to `true` so that the agent will just enforce user authentication without enforcing policies (authorization). In this integration, Oracle SSO Server handles authorization.

```
com.sun.am.policy.agents.do_sso_only = true
```

fqdnDefault

This value is set by the agent installation program to the hostname where the agent is installed. Make sure that this value is a fully qualified domain name. It should be set to *hostname.domain*. For example, if the machine is called *agent1* and the domain is *example*`.com`, this property should be set as follows:

```
com.sun.am.policy.agents.fqdnDefault = agent1.example.com
```

reverse_the_meaning_of_notenforcedList

Set this value to `true` so that the `notenforcedList` becomes the enforced list.

```
com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList = true
```

notenforcedList

Since the property `reverse_the_meaning_of_notenforcedList` is set to `true`, this property becomes the list of URLs that the agent enforces (in other words, the URLs in this list require user authentication to grant access). For example, if the fully qualified hostname of the system where the Oracle 10g Infrastructure is installed is *agent1.example.com* and the port where the Oracle SSO Server is running is *7777*, then this value should be set as follows:

```
com.sun.am.policy.agents.notenforcedList =
http://agent1.example.com:7777/pls/orasso/ORASSO.wwsso_app_admin.ls_login
http://agent1.example.com:7777/sso/auth*
```

Note that a space separates the two URLs. Additionally, the value of this property must not change based on the number or type of partner applications. It should always be set as detailed above.

logout.url

This value specifies the logout URLs of Oracle SSO Server and the partner applications. These URLs are never enforced by the agent. When the agent sees any of these URLs, it checks whether a valid session ID for the user still exists. If one does exist, the agent invalidates it, which basically logs the user out of Sun ONE Identity Server. The agent then passes the request onto Oracle SSO Server so that the logout can be processed there. In this integration, the Logout URL for Oracle SSO Server and Oracle Portal 3.0.9 are included since the Oracle Portal is the partner application that was chosen for verification. Note that these values are separated by only a space. Here is an example:

```
com.sun.am.policy.agents.logout.url =
http://hostname:port/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.logout?p_do
ne_url=http%3A%2F%2F<hostname>%3A<port>%2Fpls%2Fportal30_sso%2FPORTAL30_SS
O.home

http://hostname:port/pls/portal30/PORTAL30.wwsec_app_priv.logout?p_done_url=h
ttp%3A%2F%2F<hostname>%3A<port>%2Fpls%2Fportal30%2FPORTAL30.home
```

Here, the *hostname* refers to the host where the agent is installed and the *port* is the port of the Oracle HTTP Server. For example, if the hostname is *agent1* and the port is *7779*, then this value should be set as follows:

```
com.sun.am.policy.agents.logout.url=http://agent1:7779/pls/portal30_sso/PO
RTAL30_SSO.wwsec_app_priv.logout?p_done_url=http%3A%2F%2Fagent1%3A7779%2Fp
ls%2Fportal30_sso%2FPORTAL30_SSO.home

http://agent1:7779/pls/portal30/PORTAL30.wwsec_app_priv.logout?p_done_url=
http%3A%2F%2Fagent1%3A7779%2Fpls%2Fportal30%2FPORTAL30.home
```

NOTE If additional Oracle Partner applications are used, then the logout pages for those applications must also be added to this list.

logout.cookie_reset_list

This property lists the cookies that need to be reset or removed upon log out from Oracle SSO Server. If Oracle Portal 3.0.9 is also used for the integration, cookies for both Oracle SSO Server and Oracle Portal must be present in this list as follows:

```
com.sun.am.policy.agents.logout.cookie_reset_list = Domain=,
iPlanetDirectoryPro, iPlanetDirectoryPro;Domain=, portal30,portal30;Domain=
```

NOTE If additional Oracle partner applications are used, then cookies for those applications must also be added to this list so they can be reset/removed upon logout from Oracle SSO Server.

Verifying the Deployment

Now that the installation and configuration steps are complete, you can test the SSO integration with a partner application. Follow these steps:

1. Restart Sun ONE Identity Server. For steps, see Sun ONE Identity Server documentation.
2. Next, restart Login Server / Oracle SSO Server to ensure that all the Oracle and agent changes take place. Make sure that Login Server / Oracle SSO Server is restarted as the “oracle” user and not `root`.

   If you are deploying the SSO solution for Oracle Application Server 10g, restart the Oracle HTTP Server and OC4J_SECURITY using Enterprise Manager through the web browser.
3. Log in to Login Server / Oracle SSO Server or to the partner application, for example, Oracle Portal Server. Once you select Login, you should be directed to Sun ONE Identity Server login page. After successful authentication, you should immediately gain access to the Oracle application.

Troubleshooting Tips

- If the integration fails, look for debugging clues from the log files for Login Server and the policy agent for Apache Web Server 1.3.29, and possibly those for Sun ONE Identity Server.
- If you suspect that communication between the agent and Sun ONE Identity Server is problematic, replace the custom agent configuration file with the default version of the file, which protects all the elements on the protected port. You can then verify whether the port is working.

Chapter 6

Single Sign-On Solution for SAP Internet Transaction Server 2.0

This chapter describes the steps needed to integrate Sun ONE Identity Server into a Single Sign-On (SSO) environment with SAP's Internet Transaction Server (ITS) 2.0. Topics in this chapter include:

Introduction

SAP ITS 2.0 acts as the gateway between your web server and the backend SAP R/3 application server by adding an HTML-based user interface to SAP applications. It is composed of two parts, WGate and AGate. WGate establishes the connection between ITS and the Web server and forwards user requests to AGate, which establishes the connection to the SAP R/3 system and performs processing tasks that are required to move data between SAP R/3 applications and the Internet. WGate resides on the same computer as the web server, as a server extension.

Sun ONE Identity Server along with Sun ONE Identity Server Policy Agent provides a natural integration between the SAP applications and non-SAP applications through the use of the SAP Pluggable Authentication Service (PAS).

Architecture Details

SSO is achieved through the use of PAS provided by SAP. PAS supports several types of external authentication methods, including X.509 Certificates, NTLM, NTPassword, LDAP, HTTP and dynamic libraries (DLL). This SSO solution, using Sun ONE Identity Server, uses the DLL method for external authentication. This scenario offers SSO using a partner-specific library, which is a shared library and is developed using SAP's SDK for PAS. SAP's SDK has four functions, and provides an interface to the ITS system without the knowledge of the XGateway interface of the ITS itself.

The process flow in the SSO environment is as follows:

1. A user issues an HTTP request to a SAP service named `sapdll`.
2. The request is intercepted by the policy agent. Since there is no valid SSOToken in the request, the user is redirected to Sun ONE Identity Server for authentication.
3. Upon successful authentication, the user is granted access to the `sapdll` service. This is the PAS dynamic link library which communicates with Sun ONE Identity Server and verifies the validity of the SSO Token.
4. The PAS dynamic link library then sets the value of ~login to that of the user who authenticated with Sun ONE Identity Server and is mapped in the SAP system.
5. PAS then issues a SAP logon ticket for the user, which is set in the user's browser.
6. PAS reroutes the user to the requested service (such as Webgui).

Prerequisites

The following steps are prerequisite to ensuring that SSO scenario works properly:

- Install and configure two ITS instances. The first instance is the regular ITS which hosts the Webgui service, and the second instance is the ITS administration which hosts the PAS service.

- Configure at least one SAP system to issue SAP SSO (SSO2 logon) tickets
- Configure the other SAP systems to accept SSO2 logon tickets.
- Ensure that the browser supports and accepts cookies because SSO2 logon tickets are saved as browser cookies.
- Configure SAP Secure Network Connections (SNC) on the ticket-issuing SAP system, but not necessarily on the ticket-accepting system. SNC is a software layer in the SAP system architecture, which assures safe communication between trusted SAP components. It requires a cryptographic library to secure the data communication paths between the various SAP systems.
- Configure PAS to use an external authentication mechanism. For details, refer SAP documentation.
- Install and configure Sun ONE Identity Server and Sun ONE Identity Server Policy Agent for Sun ONE Web Server 6.0.

Installing PAS

PAS must be installed on the Administration AGate (ADM) instance. The library needed for PAS (`sapextauth.dll`) is supplied with SAP ITS from 4.6D C3 onwards, and is also located in the ITS program directory. For detailed instructions to install PAS, see SAP documentation.

The required service and template files must be installed in the respective instance in the subdirectories `\services` and `\templates`, respectively. To do so, you can unpack ITS package `ntauth.sar` from the SAP Service Marketplace or from the server component CD in path ITS\common\packages\211, or create the following files manually:

- `\services\`*`pasname`*`.srvc`
- `\templates\`*`pasname`*`\99\login.html`
- `\templates\`*`pasname`*`\99\redirect.html`
- `\templates\`*`pasname`*`\99\extautherror.html`

For details of these template files, see section SAP Template Files.

It is important to note that two separate AGate instances are required on the ticket issuing system. While PAS is installed on the ADM instance of the ticket issuing system, the Webgui service is hosted on the other AGate instance. On a ticket accepting system, only the Webgui service is hosted on the typical AGate instance.

Configuring the SAP Systems

To set up the SSO environment, you need to configure at least one SAP system to issue SSO2 logon tickets and some other systems to accept the SSO2 logon tickets. The following sections provide steps to configure these systems.

Configuring SAP R/3 System and the ITS instance

As stated in the section Prerequisites, the connection between AGate and the ticket-issuing SAP system need to be configured for SNC. The following instructions describe how to configure the SAP R/3 system and its corresponding ITS instance. For instructions on how to install SNC, please refer to the SAP SNC User's Guide.

1. On the ticket issuing SAP R/3 system, configure the following parameters in the `DEFAULT.PFL` file.

Table 6-1 Parameters in `DEFAULT.PFL`

Parameter	Value
snc/enable	1
snc/gssapi_lib	path_to_SAPCRYPTOLIB
snc/identity/as	SNC name of the application server
snc/data_protection_max	3
snc/data_protection_min	1
snc/data_protection_use	2

2. Specify AGate's SNC information in the system access control list for SNC. This list is available in the table SNCSYSACL, view VSNCSYSACL and TYPE=E.
   - Enter the SNC name for AGate in the SNC name field.
   - Select the following options:
     - Entry for RFC activated
     - Entry for diag activated

- Entry for certificate activated

3. Create a generic entry for AGate in the extended user access control list. This list is available in the table USRACLEXT:
   - Enter an asterisk (*) in the User field.
   - Enter AGate's SNC name in the SNC name field
4. If you require external user name mapping, you need to maintain the mapping in the table USREXTID.
5. In the ITS component's AGATE `global.srvc` file, configure these parameters:

Table 6-2 Parameters in `global.srvc`

Parameter	Value
~Type	2
~SncNameAgate	SNC name of AGate and ITS Manager
~sncNameR3	SNC name of the application server
~sncQoPR3	2
~secure	0

6. Make sure the environment variable `SNC_LIB` contains the path and file name of `sapcryptolib`.

Configuring the System to Issue SSO2 Logon Tickets

Use the following steps to configure the SAP R/3 Stem as well as its corresponding ITS instance for issuing SAP SSO2 logon tickets.

1. Stop the running AGate instance on the ITS server, if necessary.
2. Set the parameters in the global service file `global.srvc`:

Table 6-3 Parameters in `global.srvc`

Parameter	Value
~login	(space)

Table 6-3 Parameters in `global.srvc`

Parameter	Value
~password	(space)
~cookies	1
~mysapcomusesso2cookie	1
~mysapcomnosso1cookie	1
~mysapcomssonoits	1
~mycomgetsso2cookie	1
~secure	0
~type	2

3. Set the following parameters in the application server's profile on the ticket issuing SAP R/3 system by modifying `DEFAULT.PFL`:

Table 6-4 Parameters in `DEFAULT.PFL`

Parameter	Value
login/accept_sso2_ticket	1
login/create_sso2_ticket	2
login/ticket_expiration_time	Desired Value

4. Execute the SSO administration wizard (transaction SSO2 in the SAP system).
5. Enter NONE as the RFC destination.
6. Choose Edit->Activate Workplace.

Configuring Systems to Accept SSO2 Logon Tickets

To configure the component systems to accept and verify SSO2 logon tickets:

1. In the global service file `global.srvc`, set the following parameters:

Table 6-5 Parameters in global.srvc

Parameter	Value
~login	(space)
~password	(space)
~mysapcomusesso2cookie	1

2. On all of the component systems' application servers, set the following profile parameters:

Table 6-6 Parameters in the Component Systems' Application Servers

Parameter	Value
login/accept_sso2_ticket	1
login/create_sso2_ticket	0

3. Execute the Transaction SSO2 using the SSO administration wizard on the SAP R/3 system.
4. Enter the RFC destination or the *host name* and *system number* for the ticket issuing system.
5. If the report indicates errors on the SAP R/3 system, correct these errors on the ticket issuing SAP R/3 system and re-execute the SSO administration wizard on the component systems.
6. To initiate the configuration steps on the component system, choose Edit->Activate Workplace. Red traffic lights indicate errors in the configuration.
7. Place the PAS shared library in the `programs` directory of the SAP instance. After you have installed the policy agent, copy the policy agent shared libraries also to this directory. For details, see section Installing and Configuring the Policy Agent.

Each SAP service must have its own corresponding template files and service files. The SAPDLL service file will follow the same naming conventions as the rest of the SAP services, that is, if the service name is *sapdll*, the service file name will be *sapdll.srvc*. For more information on the template files, see section SAP Template Files.

The *sapdll.srvc* service file must be configured as follows. This file must be located under *SAP Install_dir*/SAP/ITS/2.0/ADM/services.

Table 6-7 Parameters in *sapdll.srvc*

Parameter	Value
~login	test
~password	test
~theme	99
~xgateway	sapextauth
~extauthtype	DLL
~extauthmodule	\path_to_extauth.dll
~extid_type	UN
~properties_file	\path_to_paslibrary_config_file
~exitUrl	http://s1is_host:port/amserver/UI/Logout
~client	000
~language	en
~mysapcomgetsso2cookie	1
~redirectHost	host.domain:port
~redirectPath	/scripts/wgate/webgui/!
~redirectQS	~client=000
~redirectHttps	0

NOTE The parameter ~properties_file is not a standard SAP service file parameter. This parameter should be added to the sapdll service file because the PAS DLL requires this file to know which Sun ONE Identity Server instance to communicate with.

Installing and Configuring the Policy Agent

Once you have configured the SAP R/3 systems and Sun ONE Identity Server, you can install Sun ONE Identity Server Policy Agent, version 2.1 for Sun ONE Web Server 6.0. For details on installing and configuring the policy agent, see Chapter 2 of this guide.

For the SSO solution to work properly, you must take care of the following:

- In Identity Server, policies must exist to allow or deny user access to the SAP service and resources.

  The SAP Service typically resides at:

  `http://`*`host.domain`*`:`*`port`*`/scripts/wgate/sapdll/!`

  This is the URL for the `sapdll` PAS module service, which eventually redirects the user to the requested resource as indicated by the parameters `~redirectHost` and `~redirectQS` in the `sapdll.srvc` file. Policies must exist to protect the service (`/scripts/wgate/sapdll/!`) and the corresponding redirecting resource. For information on creating policies in Sun ONE Identity Server, please see Sun ONE Identity Server documentation.

- The following policy agent shared libraries must be placed in the programs directory of your SAP ITS instance (\ *Program Files* \ SAP \ ITS \ 2.0 \ programs). For the PAS shared library to work properly, it is absolutely necessary that the shared libraries for the policy agent are accessible.

  The following are the libraries that you will need:

  - `amsdk.dll`
  - `libnspr4.dll`
  - `libplc4.dll`
  - `libplds4.dll`
  - `libxml2.dll`
  - `nss3.dll`
  - `ssl3.dll`

- The `global.srvc` file on the ITS which hosts the Webgui service must contain at least the following parameters:

Table 6-8 Required Parameters in `global.srvc`

Parameter	Value
~client	000
~cookies	1
~exiturl	http://s1is_host:port/amserver/UI/Logout
~login	(space)
~password	(space)
~xgateway	sapdiag
~xgateways	sapxgadm,sapdiag,sapxgwfc,sapxginet,sapxgbc,sapextauth
~mysapcomgetsso2cookie	1
~mysapcomusesso2cookie	1
~mysapcomnosso1cookie	1
~mysapcomssonoits	1

SAP Template Files

Along with the SAP Service file (`sapdll.srvc`), a template directory needs to be created under `ADM/templates` and it must contain the default templates. These templates are presented here. You can create these files manually at this location:

- `templates\`*pasname*`\<99>\login.html`
- `templates\`*pasname*`\<99>extautherror.html`
- `templates\`*pasname*`\<99>\redirect.html`

Template file login.html

Code Example 6-1 Template file `login.html`

```
‘declare fieldEcho, getLanguages in "sapxjutil";‘
<!--
  Copyright SAP AG 2002
  Remark: Example Login Template.
  You can write your own scripts by using BHTML and JScript
-->
‘if (~extauthtype == "LDAP")‘
```

Code Example 6-1 Template file `login.html`

```
<!--
// SAP AG 2002
// here an example for LDAP DN string. the complete string for bind must be
// uid=<user>, ou=<organisation unit>, o=<organisation>
// with this jscript example you can build your own distiguished name for
your directory
//
// This example can be used, if no Base DN is set in the service file!
// Remark: All values must not be case sensitive. After ldap_bind the module
searches
//         the correct DN in the LDAP directory and set this as ~login.
Therefore you should
//         set in USREXTID the correct DNs - USREXITD is case sensitive !
-->
<script language=javascript>
// uncomment the example code
//var ou="People";
//var o ="wdf.sap-ag.de";

function buildDN()
{
           // the input text for ~login will be replaced
   //document.pasform.elements[1].value =
"uid="+document.pasform.elements[1].value+", ou="+ou+", o="+o;
          // after new value, we submit the form --> you can see result by
jscript call
          // alert(document.pasform.elements[1].value);
   pasform.submit();
}
</script>
‘end‘
<h3>Please log on to the SAP System</h3>
<table>
   <tr><td>
   <form method="post" name="pasform" action="‘wgateURL()‘">
   ‘fieldEcho()‘
   <table>
    <tr><td>Service:</td><td>‘~Service‘</td></tr>
    ‘if (~client=="")‘
      <tr><td>Client:</td><td><input name="~client"
value="‘RSYST-MANDT‘"></td></tr>
          <tr><td>
      <input name="~clientinput" type="hidden" value= "1">
      </td></tr>
    ‘end‘</tr>
‘if (~language=="")‘<tr><td>Language:</td>
       <td>
       <select name="~language">
       ‘if (getLanguages ("langId", "langDesc") == 0)
         repeat with i from 1 to langId.dim‘
           <option value="‘langId[i]‘">‘langDesc[i]‘</option>
       ‘end
       else‘
```

Code Example 6-1 Template file `login.html`

```
        <option value="en">No allowed languages specified! Using English as
default.</option>
       ‘end‘
    </select>
       </td></tr>
    ‘end‘
<!-- for the PAS Types NTLM and HTTP the users dont have to input any things.
     for NTPassword and LDAP the Users might have to input settings like
login and password
     Remark: Administrator can predefine such things in service file like
     ~login hasso
     ~password  1972
-->
‘if (~extauthtype == "NTPassword")‘
    <tr><td>Login:</td><td><input name="~login"
value="‘~login‘"></td></tr><tr><td>
    <input name="~logininput" type="hidden" value="1">
    </td></tr>
    ‘if (~password=="")‘
    <tr><td>Password:</td><td><input type=password name="~password"
value=""></td></tr><tr><td>
    <input name="~passwdinput" type="hidden" value="1"></td></tr>
    ‘end‘
    ‘if (~extauthtype=="NTPassword")‘
<tr><td>NT domain:</td><td><input name="~ntdomain"
value="‘~ntdomain‘"></td></tr>
    ‘end‘
‘end‘

‘if (~extauthtype == "LDAP")‘
    ‘if (~login=="")‘<tr><td>Login:</td><td><input type=text name="~login"
value="‘~login‘"></td></tr><tr><td>
    <input name="~logininput" type="hidden" value="1">
    </td></tr>
    ‘end‘
    ‘if (~password=="")‘
    <tr><td>Password:</td><td><input type=password name="~password"
value=""></td></tr><tr><td>
    <input name="~passwdinput" type="hidden" value="1"></td></tr>
    ‘end‘
‘end‘

‘if (~extauthtype == "DLL")‘
    ‘if (~login=="")‘<tr><td>Login:</td><td><input type=text name="~login"
value="‘~login‘"></td></tr><tr><td>
    <input name="~logininput" type="hidden" value="1">
    </td></tr>
    ‘end‘
    ‘if (~password=="")‘
    <tr><td>Password:</td><td><input type=password name="~password"
value=""></td></tr><tr><td>
    <input name="~passwdinput" type="hidden" value="1"></td></tr>
    ‘end‘
```

Code Example 6-1 Template file login.html

```
‘end‘

<tr><td></td><td>‘~MessageLine‘</td></tr>
</table>
</td>
</tr>
<tr>
<td>
 <table align=center">
  <tr>
  <td>
  <!--
   here again for LDAP we switch the Submit button
  -->
  ‘if (~extauthtype == "LDAP")‘
  <input type=button name="~OkCode=/0"   value="Logon" onClick="buildDN()">
  ‘else‘
  <input type=submit name="~OkCode=/0"   value="Logon">
  ‘end‘
  </td>
  </tr>
 </table>
</td>
</tr>
</form>
</td>
</tr>

</table>
```

Template file extautherror.html

Code Example 6-2 Template file extautherror.html

```
<H3>Error during authentication process.</H3>

‘if (~messageline != "")‘
  <p>The following error occured:</p>‘~messageline‘
  <p> The trace files might contain more information about the problem.</p>
‘else‘
  <p>The error can’t be qualified in more detail.</p>
  <p>The trace file may contain further information about this error.</p>
‘end‘
```

Template file redirect.html

Code Example 6-3 Template file redirect.html

```
<html>
<head>
<meta http-equiv="refresh"  content="0; URL='~ExtAuthRedirectURL'">
</head>
<body>
</body>
</head>
```

Appendix A

AMAgent Properties

The configuration of Web Policy Agents is largely determined by a set of properties present in the file `AMAgent.properties`. This appendix describes the properties in this file.

NOTE All property names in this file are case-sensitive.

com.sun.am.cookieName

Description

The name of the cookie passed between Sun ONE Identity Server and the SDK.

Valid Values

The default value is iPlanetDirectoryPro. Changing the value of this property without making the corresponding change to the Identity Server will disable the SDK.

Example

```
com.sun.am.cookieName = iPlanetDirectoryPro
```

com.sun.am.namingURL

Description

The URL for the Sun ONE Identity Server Naming service.

Valid Values

The URL for the naming service

Example

`com.sun.am.namingURL =http://`*`nila.eng.example`*`.com/amserver/namingservice`

com.sun.am.policy.am.loginURL

Description

This property stores the URL of the login page on the Identity Server. If you have a failover server, you can enter its URL after a space after the primary server URL. See example.

Valid Values

URL of the login page on the Identity Server.

Example

`com.sun.am.policy.am.loginURL=http://`*`nila.eng.example.com`*`:58080/amserver/UI/Login http://`*`nila1.eng.example`*`.com:58080/amserver/UI/Login`

com.sun.am.policy.am.library.loginURL

Description

When the agent starts up, it authenticates itself as a valid agent with Sun ONE Identity Server. If a previously specified login URL must be exclusively used for redirecting users, then this property must be used to specify the Identity Server with which the agent library must authenticate.

NOTE If the value of this property is not set, the value of `com.sun.am.policy.am.loginURL` will be used as the loginURL to authenticate the agent.

Valid Values

URL of the Sun ONE Identity Server login page where the agent library must authenticate.

Example

```
com.sun.am.policy.am.library.loginURL =
http://nila.eng.example.com:58080/amserver/UI/Login
```

com.sun.am.logFile

Description

This property stores the name of the file to be used for logging messages. By default, the agent creates a directory called *webserver_instance_dir* under the /SUNWam/agents/debug directory (or the directory that you specify) and stores the log file in that directory.

Valid Values

The absolute path to the log file as shown in the following example.

Example

```
com.sun.am.logFile =/var/opt/SUNWam/agents/debug/webserver_instance_dir/amAgent
```

com.sun.am.serverLogFile

Description

This property stores the name of the Identity Server log file to be use for logging messages to Identity Server.

Valid Values

Only the file name is needed. The directory of the file is determined by settings configured on the Identity Server.

Example

```
com.sun.am.serverLogFile= amAuthLog
```

com.sun.am.logLevels

Description

This property allows to set the logging level for the specified logging categories.

Valid Values

The value of the property should be in this format:

<ModuleName>[:<Level>][,<ModuleName>[:<Level>]]*

The currently used module names are: AuthService, NamingService, PolicyService, SessionService, PolicyEngine, ServiceEngine, Notification, PolicyAgent, RemoteLog and all.

The “all” module can be used to set the logging level for all the logging modules. This will also establish the default level for all subsequently created modules.

The meaning of each level is described below:

0 Disable logging from specified module*

1 Log error messages

2 Log warning and error messages

3 Log info, warning, and error messages

4 Log debug, info, warning, and error messages

5 Like level 4, but with even more debugging messages

128 Log URL access to log file on IS server.

256 Log URL access to log file on local machine.

If level is omitted, then the logging module will be created with the default logging level, which is the logging level associated with the “all” module.

For levels 128 and 256, you must also specify a logAccessType.

Even if the level is set to zero, some messages may be produced for a module if they are logged with the special level value of “always”.

Example

```
com.sun.am.logLevels = all:5
```

com.sun.am.policy.am.username

Description

This property stores the user name to use for the Application authentication module.

Valid Values

`UrlAccessAgent`. This is a hardcoded value and must not be changed.

Examples

```
com.sun.am.policy.am.username = UrlAccessAgent
```

com.sun.am.policy.am.password

Description

This property stores the encrypted password to use for the Application authentication module.

Valid Values

The shared secret between the agent and the Identity Server it protects. By default this is the Identity Server internal LDAP authentication user (`amldapuser`) password. If you want to change this password, you must use the Shared Secret Encryption utility to do so and copy the encrypted password here. For information on the Shared Secret Encryption utility, see the section "Shared Secret Encryption Utility."

Examples

`com.sun.am.policy.am.password =` *SHARED_SECRET*

com.sun.am.sslCertDir

Description

This property stores the name of the directory containing the certificate databases for SSL.

Valid Values

The absolute path to the directory.

Example

```
com.sun.am.sslCertDir = /opt/SUNWam/servers/alias
```

com.sun.am.certDbPrefix

Description

Set this property if the certificate databases in the directory specified by the previous property have a prefix.

Valid Values

Any text string that you want to use as the prefix. This text will be prefixed to the name of the certificate databases.

Example

`com.sun.am.certDbPrefix =` *prefix*

com.sun.am.trustServerCerts

Description

This property determines if the agent should trust all server certificates when Identity Server is running SSL. For more information, see the section "The Agent's Default Trust Behavior."

Valid Values

`true` or `false`

Example

```
com.sun.am.trustServerCerts = true
```

com.sun.am.notificationEnabled

Description

This policy determines if the policy SDK should use the Identity Server notification mechanism to maintain the consistency of its internal cache. If the value is false, then a polling mechanism is used to maintain cache consistency.

Valid Values

`true` or `false`.

Example

```
com.sun.am.notificationEnabled = true
```

com.sun.am.notificationURL

Description

This property stores the URL to which notification messages should be sent if notification is enabled. See the previous property.

Valid Values

`true` or `false`

Example

```
com.sun.am.notificationURL =
http://nila.eng.example.com:58080/amagent/UpdateAgentCacheServlet?shortcir
cuit=false
```

com.sun.am.policy.am.urlComparison.caseIgnore

Description

This property determines whether case sensitivity of the URL string is obeyed during policy evaluation.

Valid Values

`true` or `false`

Example

```
com.sun.am.policy.am.urlComparison.caseIgnore = true
```

com.sun.am.policy.am.cacheEntryLifeTime

Description

This property determines the amount of time (in minutes) an entry remains valid after it has been added to the cache. The default value for this property is 3 minutes.

Valid Values

The amount of time in minutes

Example

```
com.sun.am.policy.am.cacheEntryLifeTime=3
```

com.sun.am.policy.am.userIdParam

Description

This property allows the user to configure the User Id parameter passed by the session information from the Identity Server. The value of User ID will be used by the agent to set the value of REMOTE_USER server variable. By default this parameter is set to User ID.

Valid Values

User ID

Example

```
com.sun.am.policy.am.userIdParam=UserId
```

com.sun.am.policy.am.fetchHeaders

Description

This property enables / disables the additional policy response attributes to be introduced into the HTTP headers. The value can be true or false.

Valid Values

`true` or `false`.

Example

```
com.sun.am.policy.am.fetchHeaders=false
```

com.sun.am.policy.am.headerAttributes

Description

This property determines the policy attributes to be added to the HTTP header.

NOTE In most cases, in a destination application where a “http_header_name” shows up as a request header, it will be prefixed by HTTP_, and all lower case letters will become upper case, and any - will become _;

For example, “common-name” would become “HTTP_COMMON_NAME”

Valid Values

The specification is of the format ldap_attribute_name | http_header_name[,...]. where ldap_attribute_name is the attribute in data store to be fetched and http_header_name is the name of the header to which the value needs to be assigned.

Example

```
com.sun.am.policy.am.headerAttributes=cn|common-name,ou|organizational-uni
t,o|organization,mail|email,employeenumber|employee-number,c|country
```

com.sun.am.policy.am.loadBalancer_enable

Description

This property indicates whether a load balancer is used for Identity Server services.

Valid Values

`true` or `false`.

Example

```
com.sun.am.policy.am.loadBalancer_enable = false
```

com.sun.am.policy.agents.version

Description

This property is meant for product versioning, please do not modify it.

Example

```
com.sun.am.policy.agents.version=2.1
```

com.sun.am.policy.agents.logAccessType

Description

This property allows to set the URL access logging level.

Valid Values

The allowed values are:

- `LOG_NONE` - Do not log user access to URL
- `LOG_DENY` - Log URL access that was denied.
- `LOG_ALLOW` - Log URL access that was allowed.
- `LOG_BOTH` - Log URL access that was allowed or denied.

Example

```
com.sun.am.policy.agents.logAccessType = LOG_DENY
```

com.sun.am.policy.agents.agenturiprefix

Description

The agent uses this property to support some essential functions such as notification and post-data preservation.

Valid Values

Its value should be http://*host.domain*:*port*/*agent_deployment_uri* where *host*, *domain* and *port* are FQDN and port number of the web server where the agent is installed and *agent_deployment_uri* is the URI where the web server will look for agent's related HTML pages.

Example

```
com.sun.am.policy.agents.agenturiprefix =
http://nila.eng.example.com:58080/amagent
```

com.sun.am.policy.agents.locale

Description

This property specifies the Locale setting.

Valid Values

It is recommended that you do not change this value.

Example

```
com.sun.am.policy.agents.locale = en_US
```

com.sun.am.policy.agents.instanceName

Description

This property stores the unique identifier for this agent instance.

Valid Values

This property is not currently used by the agent.

Example

```
com.sun.am.policy.agents.instanceName = unused
```

com.sun.am.policy.agents.do_sso_only

Description

This property indicates whether the agent will just enforce user authentication (SSO) without enforcing policies (authorization).

Valid Values

`true` or `false`

Example

```
com.sun.am.policy.agents.do_sso_only = false
```

com.sun.am.policy.agents.accessDeniedURL

Description

This property stores the URL of the custom page that you want to display when a user tries to access a protected resource. If no value is specified, then the agent will return an HTTP status of 403 (Forbidden).

Valid Values

The URL of the page that you want to display.

Example

```
com.sun.am.policy.agents.accessDeniedURL
=http://nila.eng.example.com:58080/urlaccessdenied.html
```

com.sun.am.policy.agents.urlRedirectParam=goto

Description

This property allows the user to configure the URL Redirect parameter for different auth modules.

Valid Values

By default this parameter is set to `goto`.

Example

```
com.sun.am.policy.agents.urlRedirectParam=goto
```

com.sun.am.policy.agents.fqdnDefault

Description

Default FQDN is the fully qualified hostname that the users should use in order to access resources on this web server instance. This is a required configuration value without which the web server may not startup correctly.

The primary purpose of specifying this property is to ensure that if the users try to access protected resources on this web server instance without specifying the FQDN in the browser URL, the agent can take corrective action and redirect the user to the URL that contains the correct FQDN.

This property is set during the agent installation and need not be modified unless absolutely necessary to accommodate deployment requirements.

WARNING: Invalid value for this property can result in the web server becoming unusable or the resources becoming inaccessible.

NOTE The property com.sun.am.policy.agents.fqdnMap provides another way by which the agent can resolve malformed access URLs used by the users and take corrective action. This property can also be used to override the behavior of the agent in cases where necessary. For example, if it is required that no corrective action such as a redirect be used for users who access the web server resources using raw IP address, you can implement this by specifying a map entry such as:

com.sun.am.policy.agents.fqdnMap = <*IP*> | <*IP*>

The agent gives precedence to the entries defined in the com.sun.am.policy.agents.fqdnMap property over the value defined in this property.

Valid Values

The fully qualified domain name of the machine where the agent is installed.

Example

com.sun.am.policy.agents.fqdnMap = *nila.Eng.example*.COM

com.sun.am.policy.agents.fqdnMap

Description

The FQDN Map is a simple map that enables the agent to take corrective action in the case where the users may have typed in an incorrect URL such as by specifying partial hostname or using an IP address to access protected resources.

Valid Values

The format for this property is:

com.sun.am.policy.agents.fqdnMap = [*invalid_hostname*|*valid_hostname*][,...]

Where *invalid_hostname* is a possible invalid hostname the user may use such as partial hostname or an IP address, and the *valid_hostname* is the corresponding valid hostname which is fully qualified. For example the following is a possible value specified for *xyz.domain1.com*:

com.sun.am.policy.agents.fqdnMap = *xyz*|*xyz.domain1*.com, *xyz.domain1*|*xyz.domain1*.com

This value maps *xyz* and *xyz.domain1* to the FQDN *xyz.domain1*.com.

At runtime, the agent refers to this map in order to take corrective actions for users who may have typed in a URL with malformed hostname. If none of the entries in this map matches the hostname specified in the user request, the agent uses the com.sun.am.policy.agents.fqdnDefault property.

WARNING: Invalid value for this property can result in the web server becoming unusable or the resources becoming inaccessible.

NOTE This property can be used for creating a mapping for more than one hostname. This may be the case when the web server protected by this agent is accessible by more than one hostname. However, the use of this feature must be done with caution as it can lead to the web server resources becoming inaccessible.

The agent gives precedence to the entries defined in this property over the value defined in com.sun.am.policy.agents.fqdnDefault property.

This property can also be used in a way that the agents use the name specified in this map instead of the web server's actual name.

Say, you want your server to be addressed as *xyz.hostname*.com whereas the actual name of the server is *abc.hostname*.com. The browser only knows *xyz.hostname*.com and you have specified policies using *xyz.hostname*.com at the Identity Server console. In this file, set the mapping as com.sun.am.policy.agents.fqdnmap = valid | *xyz.hostname*.com

com.sun.am.policy.agents.cookie_reset_enabled

Description

This property must be set to true, if this agent needs to reset cookies in the response before redirecting to Identity Server for Authentication.

Valid Value

`true` or `false`. By default this is set to `false`.

Example

```
com.sun.am.policy.agents.cookie_reset_enabled=true
```

com.sun.am.policy.agents.cookie_reset_list

Description

This property gives the comma separated list of cookies, that need to be included in the Redirect Response to Sun ONE Identity Server. This property is used only if the Cookie Reset feature is enabled.

Valid Values

The cookie details need to be specified in the following format

```
name[=value][;Domain=value]
```

If `Domain` is not specified, then the default agent domain is used to set the cookie.

Example

```
com.sun.am.policy.agents.cookie_reset_list=LtpaToken,
token=value;Domain=subdomain.domain.com
```

com.sun.am.policy.agents.cookieDomainList

Description

This property gives the space separated list of domains in which cookies have to be set in a CDSSO scenario. This property is used only if CDSSO is enabled.

Valid Values

If this property is left blank, then the fully qualified cookie domain for the agent server will be used for setting the cookie domain.

Example

```
com.sun.am.policy.agents.cookieDomainList=.example.com .madisonparc.com
```

com.sun.am.policy.agents.unauthenticatedUser

Description

This property stores the User Id to be returned if a user is accessing global allow page and is not authenticated.

Valid Values

Any user id that you want to display for the unauthenticated user.

Example

```
com.sun.am.policy.agents.unauthenticatedUser=anonymous
```

com.sun.am.policy.agents.anonRemoteUserEnabled

Description

Use this property to enable / disable REMOTE_USER processing for anonymous users.

Valid Values

`true` or `false`

Example

```
com.sun.am.policy.agents.anonRemoteUserEnabled=false
```

com.sun.am.policy.agents.notenforcedList

Description

This property stores the list of URLs for which no authentication is required.

Each service has its own not-enforced list. The service name is suffixed after `com.sun.am.policy.agents.notenforcedList`. to specify a list for a particular service. SPACE is the separator between the URLs.

Valid Values

The list of URLs for which authentication is required. Wildcards can be used to define a pattern of URLs. The URLs specified may not contain any query parameters.

Example

`com.sun.am.policy.agents.notenforcedList =`
*SERVER_PROTO://SERVER_HOST:SERVER_PORTSERVER_DEPLOY_URI/UI/**
*SERVER_PROTO://SERVER_HOST:SERVER_PORTCONSOLE_DEPLOY_URI/**

com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList

Description

This property uses a boolean attribute to indicate whether the above list is a not-enforced list or an enforced list; When the value is true, the list means enforced list, or in other words, the whole web site is open / accessible without authentication except for those URLs in the list.

Valid Values

`true` or `false`

Example

`com.sun.am.policy.agents.reverse_the_meaning_of_notenforcedList = false`

com.sun.am.policy.agents.notenforced_client_IP_address_list

Description

This property stores a list of client IP addresses. No authentication and authorization are required for the requests coming from these client IP addresses.

Valid Values

Valid IP addresses in the form of 192.168.12.2 1.1.1.1

Example

```
com.sun.am.policy.agents.notenforced_client_IP_address_list =194.164.10.2
```

com.sun.am.policy.agents.is_postdatapreserve_enabled

Description

This property enables / disables POST data preservation. By default its value is set to false.

Valid Values

`true` or `false`.

Example

```
com.sun.am.policy.agents.is_postdatapreserve_enabled = false
```

com.sun.am.policy.agents.postcacheentrylifetime

Description

This property determines the number of minutes any POST data will remain valid in the web server cache. After the specified interval, the entry will be cleared.

Valid Values

The time can be set by the Administrator. The default is 10 minutes.

Example

```
com.sun.am.policy.agents.postcacheentrylifetime = 10
```

com.sun.am.policy.agents.cdsso-enabled

Description

This property indicates if the Cross-Domain Single Sign On URL is enabled.

Valid Values

`true` or `false`

Example

`com.sun.am.policy.agents.cdsso-enabled=true`

com.sun.am.policy.agents.cdcservletURL

Description

This property indicates the URL the user will be redirected to after a successful login in a CDSSO scenario.

Valid Values

The URL to which the user will be redirected.

Example

`com.sun.am.policy.agents.cdcservletURL =`
`http://`*`sina.eng.example`*`.com:58080/amserver/cdcservlet`

com.sun.am.policy.agents.client_ip_validation_enable

Description

This property enables / disables client IP address validation. This validation will check if the subsequent browser requests come from the same IP address that the SSO token is initially issued against.

Valid Values

`true` or `false`

Example

```
com.sun.am.policy.agents.client_ip_validation_enable = false
```

com.sun.am.policy.agents.logout.url

Description

This property indicates the application's Logout URL This URL is not enforced by policy. When the agent sees this URL, it checks whether a valid session ID for the user still exists. If one does exist, the agent invalidates it, thus logging the user off Sun ONE Identity Server.

Valid Values

The logout URLs used by the protected applications.

Example

com.sun.am.policy.agents.logout.url=http://dsameqa1:7778/pls/portal30_sso/PORTAL30_SSO.wwsec_app_priv.logout?p_done_url=http%3A%2F%2Fdsameqa1%3A7778%2Fpls%2Fportal30_sso%2FPORTAL30_SSO.home

http://dsameqa1:7778/pls/portal30/PORTAL30.wwsec_app_priv.logout?p_done_url=http%3A%2F%2Fdsameqa1%3A7778%2Fpls%2Fportal30%2FPORTAL30.home

com.sun.am.policy.agents.logout.cookie_reset_list

Description

This property lists the cookies that need to be reset or removed upon log out.

Valid Values

Cookies in the same format as specified for the property com.sun.am.policy.agents.cookie_reset_list.

Example

```
com.sun.am.policy.agents.logout.cookie_reset_list =
iPlanetDirectoryPro;Domain=, iPlanetDirectoryPro;Domain=iplanet.com
```

com.sun.am.policy.am.ldapattribute.cookiePrefix

Description

If a value is specified for this field, any cookie set will have its prefix set to this value. For example, if the property is set to *MY_COOKIE_PREFIX_*, for the LDAP attribute *email*, the cookie name will be *MY_COOKIE_PREFIX_email*.This property is used when the user wants to set the LDAP attribute through a COOKIE.

Valid Values

Any string. The default value is `HTTP_`.

Example

```
com.sun.am.policy.am.ldapattribute.cookiePrefix  = HTTP_
```

com.sun.am.policy.am.ldapattribute.cookieMaxAge

Description

This property indicates the time in seconds after which the cookie will expire. This property is used when the user wants to set the LDAP attributes through COOKIE.

Valid Values

Amount of time in seconds.

Example

```
com.sun.am.policy.am.ldapattribute.cookieMaxAge = 300
```

com.sun.am.policy.agents.getClientHostname

Description

This property indicates whether to get the client's hostname through DNS reverse lookup for use in policy evaluation.

Valid Values

`true` or `false`. By default, the value is `true` if the property does not exist or if it is any value other than `false`.

Example

```
com.sun.am.policy.agents.getClientHostname = true
```

com.sun.am.policy.am.ldapattribute.mode

Description

This property is used to specify if additional policy response attributes should be introduced into the request.

Valid Values

This property takes the following values:

- NONE: indicates that no additional policy attributes will be introduced.
- HEADER: means that additional policy attributes will be introduced into HTTP header.
- COOKIE: indicates that additional policy attributes will be introduced through cookies.

  If a value other than these is supplied, it will default to NONE.

Example

```
com.sun.am.policy.am.ldapattribute.mode=COOKIE
```

com.sun.am.policy.am.fetchFromRootResource

Description

By default, when a policy decision for a resource is needed, the agent gets and caches the policy decision of the resource and all the resources from the root of the resource down, from the Identity Server.

For example, if the resource is *http://host/a/b/c*, the root of the resource is *http://host/*. This is because more resources on the same path are likely to be accessed subsequently. However, this may take a long time initially if there are many policies defined under the root resource. To have the agent get and cache the policy decision for the resource only, set this property to `false`.

NOTE This property is not currently used with Sun ONE Identity Server 6.1. It will be used with Sun ONE Identity Server 6.2 or later.

Valid Values

`true` or `false`

Example

```
com.sun.am.policy.am.fetchFromRootResource = true
```

Index

E

F

I

J

L

N

O

P

R

S

U

V

W

Appendix B

Error Codes

This appendix lists the error codes you may encounter while installing and configuring Web Policy Agents. It also provides explanation for the codes.

0. AM_SUCCESS

The operation completed successfully.

1. AM_FAILURE

The operation did not complete successfully. Please refer to the log file for more details.

2. AM_INIT_FAILURE

The C SDK initialization routine did not complete successfully. All the other APIs may be used only if the initialization went through successfully.

3. AM_AUTH_FAILURE

The authentication did not go through successfully. This error is returned either by the Authentication API or the Policy Initialization API, which tries to authenticate itself as a client to the Identity Server.

4. AM_NAMING_FAILURE

The naming query failed. Please look at the log file for further information.

5. AM_SESSION_FAILURE

The session operation did not succeed. The operation may be any of the operations provided by the session API.

6. AM_POLICY_FAILURE

The policy operation failed. Details of policy failure may be found in the log file.

7. This is a reserved error code.

8. AM_INVALID_ARGUMENT

The API was invoked with one or more invalid parameters. Check the input provided to the function.

9. This is a reserved error code.

10. This is a reserved error code.

11. AM_NO_MEMORY

The operation failed because of a memory allocation problem.

12. AM_NSPR_ERROR

The underlying NSPR layer failed. Please check log for further details.

13. This is a reserved error code.

14. AM_BUFFER_TOO_SMALL

15. AM_NO_SUCH_SERVICE_TYPE

The service type input by the user does not exist. This is a more specific version of AM_INVALID_ARGUMENT. The error may occur in any of the API that take `am_policy_t` as a parameter.

16. AM_SERVICE_NOT_AVAILABLE

17. AM_ERROR_PARSING_XML

During communication with Identity Server, there was an error while parsing the incoming XML data.

18. AM_INVALID_SESSION

The session token provided to the API was invalid. The session may have timed out or the token is corrupted.

19. AM_INVALID_ACTION_TYPE

This exception occurs during policy evaluation, if such an action type does not exist for a given policy decision appropriately found for the resource.

20. AM_ACCESS_DENIED

The user is denied access to the resource for the kind of action requested.

21. AM_HTTP_ERROR

There was an HTTP protocol error while contacting the Identity server.

22. AM_INVALID_FQDN_ACCESS

The resource provided by the user is not a fully qualified domain name. This is a web container specific error and may be returned by the `am_web_is_access_allowed` function only.

23. AM_FEATURE_UNSUPPORTED

The feature being invoked is not implemented as of now. Only the interfaces have been defined.

24. AM_AUTH_CTX_INIT_FAILURE

The Auth context creation failed. This error is thrown by `am_auth_create_auth_context`.

25. AM_SERVICE_NOT_INITIALIZED

The service is not initialized. This error is thrown by `am_policy` functions if the provided service was not initialized previously using `am_policy_service_init`.

26. AM_INVALID_RESOURCE_FORMAT

This is a plugin interface error. Implementors of the new resource format may throw this error if the input string does not meet their specified format. This error is thrown by the `am_web` layer, if the resource passed as parameter does not follow the standard URL format.

27. AM_NOTIF_NOT_ENABLED

This error is thrown if the notification registration API is invoked when the notification feature is disabled in the configuration file.

28. AM_ERROR_DISPATCH_LISTENER

Error during notification registration.

29. AM_REMOTE_LOG_FAILURE

This error code indicates that the service that logs messages to Sun ONE Identity Server has failed. The details of this error can be found in the agent's log file.