Sun Java System Access Manager Policy Agent 2.2 Guide for Apache HTTP Server 2.0.54

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

Part No: 819–4770–14
January 21, 2008

080201@19028

Contents

Preface

This Sun Java System Access Manager Policy Agent 2.2 Guide for Apache HTTP Server 2.0.54 is a web agent guide. Therefore, it provides general information about web agents in the Sun Java™ System Access Manager Policy Agent 2.2 software set. This guide also provides specific information about Sun Java System Access Manager Policy Agent for Apache HTTP Server. This web agent does not only support Apache HTTP Server version 2.0.54. It also supports version 1.3.33. For more support and compatibility information, see "Supported Platforms of Agent for Apache HTTP Server" on page 25.

Included in this guide is information about installing, configuring, uninstalling, and troubleshooting web agents, with the focus being on Policy Agent for Apache HTTP Server.

Who Should Use This Book

This *Sun Java System Access Manager Policy Agent 2.2 Guide for Apache HTTP Server 2.0.54* is intended for use by IT professionals who manage access to their network using Sun Java System servers and software. Administrators should understand the following technologies:

- Directory technologies
- JavaServer Pages™ (JSP) technology
- HyperText Transfer Protocol (HTTP)
- HyperText Markup Language (HTML)
- eXtensible Markup Language (XML)
- Web Services
- Web Technologies

Before You Read This Book

Sun Java System Policy Agent software works with Sun Java System Access Manager. Both products work with Sun Java Enterprise System, a software infrastructure that supports enterprise applications distributed across a network or Internet environment. Furthermore, Sun Java System Directory Server is a necessary component in a new Access Manager deployment since it is used as the data store. To understand how these products interact and to understand this book, you should be familiar with the following documentation:

- Sun Java Enterprise System documentation set, which can be accessed online at `http://docs.sun.com`. All Sun technical documentation is available online through this web site, including the other documentation sets referred to in this list.

  You can browse the documentation archive or search for a specific book title, part number, or subject.
- Sun Java System Directory Server documentation set.
- Sun Java System Access Manager documentation set, which is explained in more detail subsequently in this chapter.
- Sun Java System Access Manager Policy Agent 2.2 documentation set, which is explained in more detail subsequently in this chapter.

How This Book Is Organized

This book is organized in the following manner:

Preface, this chapter, provides information about this book to help you use the book to your best advantage.

Chapter 1, "Introduction to Web Agents for Policy Agent 2.2," introduces web agents in Policy Agent 2.2, focusing on what all web agents have in common in this release.

Chapter 2, "About Policy Agent 2.2 for Apache HTTP Server," provides information specific to Policy Agent 2.2 for Apache HTTP Server, focusing on aspects of the agent that make it unique compared to other web agents.

Chapter 3, "Installing Policy Agent 2.2 for Apache HTTP Server," provides instructions for installing Policy Agent 2.2 for Apache HTTP Server.

Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2," provides information about the agent profile, which is an optional location for setting the credentials that the web agent must provide to authenticate with Access Manager.

Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for Apache HTTP Server," provides information about web agent configuration.

Chapter 6, "Managing Policy Agent 2.2 for Apache HTTP Server," provides information about the methods available for managing Policy Agent 2.2 for Apache HTTP Server, with most of the information being applicable to all web agents in the Policy Agent 2.2 software set.

Chapter 7, "Uninstalling Policy Agent 2.2 for Apache HTTP Server," provides instructions for uninstalling Policy Agent 2.2 for Apache HTTP Server.

Appendix A, "Silent Installation of a Web Agent in Policy Agent 2.2" provides instructions for creating and using a script for automatic installation of a web agent in the Policy Agent 2.2 software set.

Appendix B, "Troubleshooting a Web Agent Deployment" provides troubleshooting instructions for problems that might occur in Policy Agent 2.2 for Apache HTTP Server.

Appendix C, "Web Agent `AMAgent.properties` Configuration File" provides a list of the properties in the web agent `AMAgent.properties` configuration file in Policy Agent 2.2 for Apache HTTP Server, with most properties being applicable to all the web agents in the Policy Agent 2.2 software set.

Appendix D, "Error Codes" provides a list of error codes that might be encountered during installation or configuration.

Related Books

Sun Microsystems server documentation sets, some of which are mentioned in this preface, are available at `http://docs.sun.com`. These documentation sets provide information that can be helpful for a deployment that includes Policy Agent.

Access Manager Documentation Set

Policy Agent 2.2 was first introduced with Access Manager 7, but now also supports Access Manager 7.1. The information in the table that follows specifies documents in the Access Manager 7 documentation set, which is available at the following location:

`http://docs.sun.com/app/docs/coll/1292.1`

The Access Manager 7.1 documentation set is available at this location:

`http://docs.sun.com/app/docs/coll/1292.2`

TABLE P–1 Access Manager 7 2005Q4 Documentation Set

Title	Description
Sun Java System Access Manager 7 2005Q4 Release Notes	Available after the product is released. Contains last-minute information, including a description of what is new in this current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.
Sun Java System Access Manager 7 2005Q4 Technical Overview	Provides an overview of how Access Manager components work together to consolidate identity management and to protect enterprise assets and web-based applications. Explains basic Access Manager concepts and terminology

TABLE P–1 Access Manager 7 2005Q4 Documentation Set *(Continued)*

Title	Description
Sun Java System Access Manager 7 2005Q4 Deployment Planning Guide	Provides information about planning a deployment within an existing information technology infrastructure
Sun Java System Access Manager 7 2005Q4 Performance Tuning Guide	Describes how to tune Access Manager and its related components.
Sun Java System Access Manager 7 2005Q4 Administration Guide	Describes how to use the Access Manager console as well as how to manage user and service data via the command line.
Sun Java System Access Manager 7 2005Q4 Federation and SAML Administration Guide	Provides information about the features in Access Manager that are based on the Liberty Alliance Project and SAML specifications. It includes information on the integrated services based on these specifications, instructions for enabling a Liberty-based environment, and summaries of the application programming interface (API) for extending the framework.
Sun Java System Access Manager 7 2005Q4 Developer's Guide	Offers information on how to customize Access Manager and integrate its functionality into an organization's current technical infrastructure. Contains details about the programmatic aspects of the product and its API.
Sun Java System Access Manager 7 2005Q4 C API Reference	Provides summaries of data types, structures, and functions that make up the Access Manager public C APIs.
Sun Java System Access Manager 7 2005Q4 Java API Reference	Are generated from Java code using the JavaDoc tool. The pages provide information on the implementation of the Java packages in Access Manager.
Sun Java System Access Manager Policy Agent 2.2 User's Guide	Provides an overview of Policy Agent software, introducing web agents and J2EE agents. Also provides a list of web agents and J2EE agents currently available.

Updates to the *Release Notes* and links to modifications of the core documentation can be found on the Access Manager page at the Sun Java System 2005Q4 documentation web site. Updated documents are marked with a revision date.

Policy Agent 2.2 Documentation Set

Other Policy Agent guides, besides this guide, are available as described in the following sections:

Sun Java System Access Manager Policy Agent 2.2 User's Guide

The *Sun Java System Access Manager Policy Agent 2.2 User's Guide* is available in two documentation sets: the Access Manager documentation set as described in Table P–1 and in the Policy Agent 2.2 documentation set as described in this section.

Other Individual Agent Guides

The individual agents in the Policy Agent 2.2 software set, of which this book is an example, are available on a different schedule than Access Manager itself. Therefore, documentation for Access Manager and Policy Agent are available in separate sets, except for the *Sun Java System Access Manager Policy Agent 2.2 User's Guide*, which is available in both documentation sets.

The documentation for the individual agents is divided into two subsets: a web Policy Agent subset and a J2EE Policy Agent subset.

Each web Policy Agent 2.2 guide provides general information about web agents and installation, configuration, and uninstallation information for a specific web agent.

Each J2EE Policy Agent 2.2 guide provides general information about J2EE agents and installation, configuration, and uninstallation information for a specific J2EE agent.

The individual agent guides are listed along with supported server information in the following chapters of the *Sun Java System Access Manager Policy Agent 2.2 User's Guide*:

Web Agents	Chapter 2, "Access Manager Policy Agent 2.2 Web Agents: Compatibility, Supported Servers, and Documentation," in *Sun Java System Access Manager Policy Agent 2.2 User's Guide*
J2EE Agents	Chapter 3, "Access Manager Policy Agent 2.2 J2EE Agents: Compatibility, Supported Servers, and Documentation," in *Sun Java System Access Manager Policy Agent 2.2 User's Guide*

Release Notes

The *Sun Java System Access Manager Policy Agent 2.2 Release Notes* are available online after an agent or set of agents is released. The release notes include a description of what is new in the current release, known problems and limitations, installation notes, and how to report issues with the software or the documentation.

Sun Java Enterprise System Product Documentation

For useful information for related products, see the following documentation collections on the Sun Java Enterprise System documentation web site (`http://docs.sun.com/prod/entsys.05q4`)

- Sun Java System Directory Server:

  `http://docs.sun.com/coll/1316.1`
- Sun Java System Web Server:

  `http://docs.sun.com/coll/1308.1`
- Sun Java System Application Server:

  `http://docs.sun.com/coll/1310.1`
- Sun Java System Message Queue:

  `http://docs.sun.com/coll/1307.1`
- Sun Java System Web Proxy Server:

  `http://docs.sun.com/coll/1311.1`

Accessing Sun Resources Online

For product downloads, professional services, patches and support, and additional developer information, go to the following:

Download Center
`http://wwws.sun.com/software/download`

Sun Java System Services Suite
`http://www.sun.com/service/sunps/sunone/index.html`

Sun Enterprise Services, Solaris Patches, and Support
`http://sunsolve.sun.com/`

Developer Information
`http://developers.sun.com/prodtech/index.html`

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in the product documentation, go to:

http://www.sun.com/service/contacting

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 or alleged to be caused by or in connection with use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions.

To share your comments, go to http://docs.sun.com and click Send Comments. In the online form, provide the document title and part number. The part number is a seven-digit or nine-digit number that can be found on the title page of the guide or at the top of the document.

For example, the title of this guide is *Sun Java System Access Manager Policy Agent 2.2 Guide for Apache HTTP Server 2.0.54*, and the part number is 819-4770.

Documentation, Support, and Training

Sun Function	URL	Description
Documentation	http://www.sun.com/documentation/	Download PDF and HTML documents, and order printed documents
Support and Training	http://www.sun.com/training/	Obtain technical support, download patches, and learn about Sun courses

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–2 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. Perform a *patch analysis*. Do *not* save the file. [Note that some emphasized items appear bold online.]

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–3 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

◆ ◆ ◆ CHAPTER 1

Introduction to Web Agents for Policy Agent 2.2

The Sun Java™ System Access Manager Policy Agent 2.2 software set includes J2EE agents and web agents. This guide discusses web agents, the functionality of which has increased for this release. This chapter provides a brief overview of web agents in the 2.2 release as well as some concepts you need to understand before proceeding with a web agent deployment. For a general introduction of agents, both J2EE agents and web agents, see *Sun Java System Access Manager Policy Agent 2.2 User's Guide*.

Topics in this chapter include:

Uses of Web Agents

Web agents function with Sun Java System Access Manager to protect content on deployment containers, such as 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. Web agents perform these tasks while providing single sign-on (SSO) and cross domain single sign-on (CDSSO) capabilities as well as URL protection.

Web agents are installed on deployment containers for a variety of reasons. Here are three examples:

- A web agent on a human resources server prevents non-human resources personnel from viewing confidential salary information and other sensitive data.
- A web agent on an operations deployment container allows only network administrators to view network status reports or to modify network administration records.

- A web agent on an engineering deployment container allows authorized personnel from many internal segments of a company to publish and share research and development information. At the same time, the web 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 deployment container. For information on setting policies and for assigning roles and policies to users, see the *Sun Java System Access Manager 7 2005Q4 Administration Guide*.

How Web Agents Work

When a user points a browser to a particular URL on a protected deployment container, a variety of interactions take place as explained in the following numbered list. See the terminology list immediately following this numbered list for a description of terms.

1. The web agent intercepts the request and checks information in the request against not-enforced lists. If specific criteria are met, the authentication process is by passed and access is granted to the resource.
2. If authentication is required, the web agent validates the existing authentication credentials. If the existing authentication level is insufficient, the appropriate Access Manager Authentication Service will present a login page. The login page prompts the user for credentials such as username and password.
3. 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 Java System 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.
4. If the user's credentials are properly authenticated, the web agent checks if the users is authorized to access the resource.
5. Based on the aggregate of all policies assigned to the user, the individual is either allowed or denied access to the URL.

Terminology: How Web Agents Work

Authentication Level	The ability to access resources can be divided into levels. Therefore, different resources on a deployment container (such as a web server or a proxy server) might require different levels of authentication
Service	Access Manager is made of many components. A service is a certain type of component that performs specific tasks. Some of the Access Manager services available are Authentication Service, Naming Service, Session Service, Logging Service, and Policy Service.

Authentication Module	An authentication interface, also referred to as an authentication module, is used to authenticate a user on Access Manager.
Roles	Roles are a Directory Server entry mechanism. A role's members are LDAP entries that possess the role.
Policy	A policy defines rules that specify access privileges to protected resources on a deployment container, such as a web server.

What's New About Web Agents

Several important features have been added to the web agents in the 2.2 release as follows:

Support for Fetching User Session Attributes

Before this release of web agents, header and cookie information was retrieved, or *sourced*, solely from user profile properties. Now, header and cookie information can also be sourced from session properties.

Use the following property to choose how you want session attributes retrieved:

```
com.sun.am.policy.agents.config.session.attribute.fetch.mode
```

For the preceding property, the following modes are available as retrieval methods:

- NONE
- HTTP_HEADER
- HTTP_COOKIE

The following example illustrates this property with the retrieval method set to `HTTP_HEADER`:

```
com.sun.am.policy.agents.config.session.attribute.fetch.mode = HTTP_HEADER
```

The source of header and cookie information is controlled by the following configuration property in the web agent `AMAgent.properties` configuration file:

```
com.sun.am.policy.agents.config.session.attribute.map
```

This configuration property has the same format as an LDAP header property. The following is an example of how this configuration property can be set:

```
com.sun.am.policy.agents.config.session.attribute.map =
name-of-session-attribute1|name-of-header-attribute1,
name-of-session-attribute2|name-of-header-attribute2
```

Where *name-of-session-attribute1* and other similarly named properties, or *attributes*, in the preceding code represent actual property names.

Benefit - Support for Fetching User Session Attributes: The benefit of this feature is that session properties can be more effective for transferring information, especially dynamic information. Prior to this release, agents could only fetch users' profile attributes, which tend to be static attributes. However, session attributes allow applications to obtain dynamic user information when necessary. Since this feature allows you to fetch non-user profile attributes, you can fetch attributes such as SAML assertion.

Log Rotation

Note – Log rotation is not supported on Policy Agent 2.2 for Apache HTTP Server. For more information see "Information Specific to Agent for Apache HTTP Server" on page 27

Policy-Based Response Attributes

Starting with this release of web agents, a new method is available for retrieving LDAP user attributes based on Access Manager policy configurations.

Policy-based response attributes take advantage of functionality now available in Access Manager that involves querying policy decisions. In previous versions of Access Manager, header attributes could only be determined by the list of attribute-value pairs in the agent configuration. Now, header attributes can also be determined by Access Manager policy configurations. With policy-based response attributes you can define attribute-value pairs at each policy definition as opposed to the method used in prior versions of Access Manager, which only allowed pairs to be defined globally in the agent configuration. For more information on policy-based response attributes, see "Providing Personalization With Policy-Based Response Attributes" on page 88

Benefit - Policy-Based Response Attributes: The benefit of policy-based response attributes is that they allow for personalization, improve the deployment process, allow greater flexibility in terms of customization, and provide central and hierarchical control of attribute values.

Personalization is provided in that an application can retrieve specific user information, such as a name, from a cookie or HTTP header and present it to the user in the browser.

Defining attribute-value pairs at each policy definition instead of at the root level allows an attribute value to be distributed only to the applications that need it. Furthermore, you can customize attribute names allowing the same attribute name to have entirely different property values for two different applications.

Composite Advice

Starting with this release, web agents provide a composite advice feature. This feature allows the policy and authentication services of Access Manager to decouple the advice handling mechanism of the agents. This allows you to introduce and manage custom advices by solely writing Access Manager side plug-ins. Starting with this release, you are not required to make changes on the agent side. Such advices are honored automatically by the composite advice handling mechanism.

Benefit - Composite Advice: A benefit of composite advice is that you can incorporate a custom advice type without having to make changes to an agent deployment. Prior to the 2.2 release of web agents, no interface existed on the client side to write client-side plug-ins.

Additional Method for Fetching the `REMOTE_USER` Server Variable

Prior to this release of web agents, the only method for fetching the value of the `REMOTE_USER` variable set by an agent was from session properties. Starting with the 2.2 release, the value can also be fetched from user profiles. This fetching process uses LDAP.

By default the value for the `REMOTE_USER` is fetched from the session. If the value needs to be fetched from LDAP, the following property needs to be defined in the web agent `AMAgent.properties` configuration file:

```
com.sun.am.policy.am.userid.param.type = LDAP
```

The following property can still be used to configure the key (*key* refers to the value assigned to this property) that needs to be searched. In addition to setting the preceding property, you need to give the correct LDAP attribute name for the following property.

```
com.sun.am.policy.am.userid.param
```

For example the property will be set as follows:

```
com.sun.am.policy.am.userid.param = ldap-attribute-name
```

where *ldap-attribute-name* represents the name of an LDAP attribute.

To enable the `REMOTE_USER` setting for a globally not-enforced URL as specified in the web agent `AMAgent.properties` configuration file (this is a URL that can be accessed by unauthenticated users) you must set the following property in the web agent `AMAgent.properties` configuration file to `true`. While the following example, has the value is set to `true`, the default value is `false`:

```
com.sun.am.policy.agents.config.anonymous_user.enable = 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 web agent `AMAgent.properties` configuration file. In the following example the value is set to `anonymous`, which is the default:

```
com.sun.am.policy.agents.config.anonymous_user = anonymous
```

Benefit - Additional Method for Fetching the REMOTE_USER Server Variable: The benefit of this feature is that it gives better customization for end users since the `REMOTE_USER` server variable can now be obtained from either session attributes or user profile attributes.

Also, you do not need to write server-side plug-in code in order to add session attributes after authentication, which is necessary when this value is fetched from session properties.

Malicious Header Attributes Automatically Cleared by Agents

Starting with this release of web agents, malicious header attributes are automatically cleared.

Benefit - Header Attributes Set by Agents Automatically Cleared: The benefit of this automatic clean up is that security is improved. Header information that is *not* automatically cleared has greater risk of being accessed.

Load Balancing Enablement

Starting with this release of web agents, the default agent host port and protocol settings can be overridden to enable load balancing. For more information, see "Enabling Load Balancing" on page 96.

Benefit - Load Balancing Enablement: The benefit of this override capability is that you do not need to manually change the hostname, port, and protocol settings to enable load balancing.

Support for Heterogeneous Agent Types on the Same Machine

Starting with this release of web agents, you can install different types of agents on the same machine. Prior to this release, you could not install web agents from different product groups on the same machine. For example, previously, an agent instance for Apache HTTP Server and an agent instance for Sun Java System Web Server 6.1 could not be installed on the same machine. Now, they can.

Benefit - Support for Heterogeneous Agent Types on Same Machine: The benefit of this feature is that a deployment that has agents in a multi-server scenario requires fewer hardware sources.

Support for Turning Off FQDN Mapping

Starting with this release, fully qualified domain name (FQDN) mapping of HTTP requests can be disabled. In prior web agent releases, the methods employed for checking if a user is using a valid URL could not be turned off.

This checking capability is controlled by the FQDN default and the FQDN map properties in the web agent `AMAgent.properties` configuration file as follows:

- `com.sun.am.policy.agents.config.fqdn.default`
- `com.sun.am.policy.agents.config.fqdn.map`

A toggling capability has been introduced that allows FQDN checking to be turned off. The following property allows for this toggling:

```
com.sun.am.policy.agents.config.fqdn.check.enable
```

The following property specifies whether the request URLs that are present in user requests are checked against the FQDN default and the FQDN map properties by the web agent:

```
com.sun.am.policy.agents.config.fqdn.check.enable
```

The valid values are `true` and `false`.

`true` The request URLs that are present in user requests are checked against FQDN values.

`false` No checking occurs against FQDN values.

The default value is `true`. If no value is specified, then the default value, `true`, is used.

Benefit - Support for Turning Off FQDN Mapping: This feature allows you to turn off or on FQDN mapping comparison. This feature can be beneficial when a deployment includes a number of virtual servers for which the agent is configured using FQDN mapping.

Backward Compatibility With Access Manager 6.3

Policy Agent 2.2 is backward compatible with Access Manager 6.3 Patch 1 or greater.

Note – Policy Agent 2.2 is only compatible with Access Manager 6.3 when the Access Manager patch has been applied.

Be aware that Policy Agent 2.2 takes advantage of certain features that exist in Access Manager 7 that do not exist in Access Manager 6.3, such as "composite advices," "policy-based response attributes," and others.

◆ ◆ ◆ CHAPTER 2

About Policy Agent 2.2 for Apache HTTP Server

This chapter provides information about Sun Java System Policy Agent 2.2 as it pertains specifically to Apache HTTP Server.

While the individual web agents tend to be similar in terms of installation and configuration, they can have unique characteristics that allow them to interact with unique characteristics in the underlying deployment container, such as a web server or proxy server. Therefore, this chapter describes characteristics that are unique to this agent, Sun Java SystemAccess Manager Policy Agent 2.2 for Apache HTTP Server, and that are unique to just the deployment container, Apache HTTP Server. This chapter also summarizes specific tasks you might need to perform because of the unique characteristics of the deployment container.

Supported Platforms and Compatibility of Agent for Apache HTTP Server

The following sections provide information about the supported platforms of Policy Agent 2.2 for Apache HTTP Server as well as the compatibility of this agent with Access Manager.

Supported Platforms of Agent for Apache HTTP Server

The following table presents the supported platforms of Policy Agent 2.2 for Apache HTTP Server. Notice that this document describes what is technically two separate agents: Agent for Apache HTTP Server 1.3.33 and Agent for Apache HTTP Server 2.0.54. Throughout this guide, these two agents are referred to together as Agent for Apache HTTP Server.

This guide does not apply to Agent for Apache HTTP Server 2.2. For information about that agent, see *Sun Java System Access Manager Policy Agent 2.2 Guide for Apache HTTP Server 2.2*.

The supported platforms for these two agents differ as shown in the table that follows.

TABLE 2–1 Supported Platforms of Agent for Apache HTTP Server

Agent for	Supported Platforms
Apache HTTP Server 1.3.33* **Note –** Be aware that this guide is applicable to both Agent for Apache HTTP Server 1.3.33 and Agent for Apache HTTP Server 2.0.54	Solaris™ Operating System (OS) for the SPARC® platform, versions 8, 9, and 10 Solaris (OS) for x86 platforms, versions 8, 9, and 10 Red Hat Enterprise Linux Advanced Server 3.0 SUSE Linux Enterprise Server 9
Agent for	**Supported Platforms**
Apache HTTP Server 2.0.54*	Solaris Operating System (OS) for the SPARC platform, versions 8, 9, and 10 Solaris (OS) for x86 platforms, versions 8, 9, and 10 AIX 5L, versions 5.1, 5.2, and 5.3 Red Hat Enterprise Linux Advanced Server 3.0, versions 32 bit and 64 bit Red Hat Enterprise Linux Advanced Server 4.0, versions 32 bit and 64 bit SUSE Linux Enterprise Server 9 Debian GNU/Linux 3.0 Windows 2003, Enterprise Edition Windows 2003, Standard Edition

*Agent for Apache HTTP Server has been verified with the Apache default Multi-Processing Mode (MPM) on all supported platforms as described in `http://httpd.apache.org/docs/2.0/mpm.html`.

Compatibility of Agent for Sun Java System Apache HTTP Server With Access Manager

All agents in the Policy Agent 2.2 release are compatible with versions of Sun Java System Access Manager as described in this section.

Compatibility of Policy Agent 2.2 With Access Manager 7 and Access Manager 7.1

All agents in the Policy Agent 2.2 release are compatible with Access Manager 7 and Access Manager 7.1. Compatibility applies to both of the available modes of Access Manager: Realm Mode and Legacy Mode.

Install the latest Access Manager patches to ensure that all enhancements and fixes are applied. For an example of Access Manager patches that can be installed, see the compatibility information discussed in *Sun Java System Access Manager Policy Agent 2.2 Release Notes*.

Compatibility of Policy Agent 2.2 With Access Manager 6.3

All agents in Policy Agent 2.2 are also compatible with Access Manager 6.3 Patch 1 or greater. However, certain limitations apply. For more information about the limitations, see "Backward Compatibility With Access Manager 6.3" on page 24.

Information Specific to Agent for Apache HTTP Server

This section describes anything that is unique about this specific web agent.

Policy Agent 2.2 for Apache HTTP Server is unique in that it does not support the following features:

Notifications

Sun Java System Policy Agent 2.2 for Apache HTTP Server does not support notifications. Therefore, updating the cache through a notification mechanism is not an available feature. However, since the notification mechanism is available for other agents in the Policy Agent 2.2 software set, a property exists in the web agent `AMAgent.properties` configuration file. The property that controls the notification mechanism, `com.sun.am.notification.enable`, is set to `false` for this agent. Do not set this property to true for this agent as it might result in unexpected behavior. The two following properties can also affect notifications for most agents:

```
com.sun.am.notification.url
override_notification.url
```

However, you can ignore these properties for this agent.

Log Rotation

The multi-process environment of Apache HTTP Server impairs the agent's ability to obtain the correct size of the log file. Without a correct reading of the log file size, the agent cannot rotate the file as intended.

While most web agents have log rotation turned on by default, this agent has log rotation turned off by default. Log rotation is controlled by a property in the web agent `AMAgent.properties` configuration file. The following example illustrates the default setting of this property for this agent:

```
com.sun.am.policy.agents.config.local.log.rotate = false
```

Do not change the value of this property to `true` for this agent. A setting of `true` results in log rotation that is inconsistent and unpredictable.

SUSE Linux Enterprise 9: Libraries Needed

Agent for Apache HTTP Server is dependent upon a variety of libraries that are not included with SUSE Linux Enterprise Server 9. Therefore, if you are using a SUSE Linux system, you must perform a post-installation task to make these libraries available. See "Agent for Apache HTTP Server on SUSE Linux: Obtaining the Required Libraries" on page 73.

◆ ◆ ◆ CHAPTER 3

Installing Policy Agent 2.2 for Apache HTTP Server

Policy Agent 2.2 works in tandem with Access Manager to control user access to deployment containers (such as web servers) in an enterprise.

This chapter explains how to install Policy Agent 2.2 for Apache HTTP Server on the supported platforms. For more information on the supported platforms, see "Supported Platforms and Compatibility of Agent for Apache HTTP Server" on page 25.

For this chapter, each platform-related section leads you through the pre—installation and installation steps. First, perform the pre-installation (preparation) steps. Then, perform the installation, itself. After you complete the installation, verify that the installation was successful.

Next, complete the required post-installation tasks described in Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for Apache HTTP Server."

In this chapter, the section about verifying a successful installation describes a task that applies to all platform types. Each of the other sections of this chapter focuses on installing Apache HTTP Server on a specific platform type. The sections are as follows:

Solaris Systems: Agent Installation for Apache HTTP Server

This section describes the installation process on Solaris systems.

Preparing to Install Agent for Apache HTTP Server on Solaris Systems

Caution – Do not use the version of Apache HTTP Server that comes bundled with Solaris™ 9 Operating System or with Solaris 10 Operating System. The bundled Apache HTTP Server package is incomplete. Any attempt to Install Agent for Apache HTTP Server on a bundled version of Apache HTTP Server is likely to fail.

Therefore, download the desired version of Apache HTTP Server from the Apache web site at `http://www.apache.org/` before attempting to install the agent.

Follow the specific steps outlined in this section before you install the web agent to reduce the chance of complications occurring during and after the installation.

▼ To Prepare to Install Policy Agent 2.2 for Apache HTTP Server on Solaris Systems

Note – 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 web agent installation program. Currently, JRE 1.3.1 or any version higher is certified for use with the web agent installation program.

Perform the following pre-installation tasks:

1. **Ensure that Policy Agent 2.2 for Apache HTTP Server is supported on the desired platform as listed in "Supported Platforms and Compatibility of Agent for Apache HTTP Server" on page 25.**

2. **Install Apache HTTP Server if not already installed.**

   Refer to the Apache HTTP Server documentation for details on how best to install and configure this server for your platform.

3. **Ensure that Apache HTTP Server has the latest patches available.**

4. **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:
   ```

Installing Agent for Apache HTTP Server on Solaris Systems

The web agent installation program has two interfaces: the graphical user interface (GUI) and the command-line interface. The following sections present instructions to install the web agent using both of these interfaces:

- "GUI Installation of Agent for Apache HTTP Server on Solaris Systems" on page 31
- "Command-Line Installation of Agent for Apache HTTP Server on Solaris Systems" on page 34

GUI Installation of Agent for Apache HTTP Server on Solaris Systems

Use the following instructions to install a web agent using the GUI on Solaris systems.

▼ To Install Agent for Apache HTTP Server on Solaris Systems Using the GUI

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

1. **Unpack the product binary in the directory of your choice using the following command:**

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

2. **In the directory in which you unpack the binaries, issue the following command:**

   ```
   # ./setup
   ```

   The Welcome page appears.

3. **In the Welcome page, click Next.**

4. **Read the License Agreement. Click Yes to agree to the license terms.**

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

   Install the web agent in this directory: Enter the full path to the directory where you want to install the web agent. The default installation directory is `/opt`.

   The directory you choose in which to install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

6. **Click Next and provide the following information about the Apache HTTP Server instance the agent will protect:**

   Host Name: Enter the fully qualified domain name (FQDN) of the machine where the Apache HTTP Server instance is installed.

For example, if the host is `host1`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host1.eng.example.com`.

Apache Binary Directory: Enter the full path to the directory where the Apache HTTP Server binary, therefore the `httpd` binary, is installed. An example pathname follows:

Apache-base`/bin`

where *Apache-base* represents the directory where Apache HTTP Server was installed. Refer to the Apache HTTP Server documentation for the specific path name.

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

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

Agent Deployment URI: Enter a Universal Resource Identifier (URI) that will be used to access Agent for Apache HTTP Server. The default value is `/amagent`.

Note – The web agent uses the value of the `com.sun.am.policy.agents.config.agenturi.prefix` property in the web agent `AMAgent.properties` configuration file to support some essential functions such as notification. 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 Apache HTTP Server instance where the agent is installed and *agent-deployment-uri* is the URI where the Apache HTTP Server instance will look for web-agent related HTML pages. Its default value is `amagent`.

The following is an example of an Agent Deployment URI:

```
http://host1.example.com:80/amagent
```

Apache Config Directory: Enter the full path to the directory that contains the Apache HTTP Server configuration file `httpd.conf`. An example pathname follows:

Apache-base`/conf`

where *Apache-base* represents the directory where Apache HTTP Server was installed.

SSL Ready: Select this option if the Apache HTTP Server instance you are using has support for SSL. Your Apache HTTP Server instance 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 HTTP Server instance has been compiled with the EAPI flag, go to the `bin` directory of the Apache HTTP Server instance and type the following command:

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

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

The supported configurations for Apache HTTP Server are:

- Apache HTTP Server without `mod_ssl` support
- Apache HTTP Server with `mod_ssl` and EAPI flag enabled.

Note – Apache HTTP Server with `mod_ssl` support and EAPI flag disabled configuration is not supported by Policy Agent 2.2.

---

7 When you have entered all the information correctly, click Next.

8 Enter information about the Access Manager host.

The web agent will connect to this server.

Primary Server Host: Enter the FQDN of the primary Access Manager host.

For example, if the host is `host3`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host3.eng.example.com`.

Primary Server Port: Enter the port number for the primary Access Manager host.

Primary Server Protocol: If the primary Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP.

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

Primary Console Deployment URI: Enter the location that was specified when Access Manager Console was installed. The default URI for Access Manager is `/amconsole`.

Failover Server Host: Enter the FQDN of the secondary Access Manager host if the primary Access Manager host becomes unavailable. If no failover server host exists, then leave this field blank.

Failover Server Port: Enter the port number of the secondary Access Manager host. If no failover server host exists, then leave this field blank.

Failover Server Protocol: If the failover Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP. If no failover server host exists, then leave this field blank.

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

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

Agent Access Manager Shared Secret: Enter the password for the Access Manager internal LDAP authentication user. This user is also referred to as `amldapuser`.

For more information about the shared secret and its relationship with the Access Manager agent profile, see Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2."

Re-enter Shared Secret: Re-enter the password for the Access Manager internal LDAP authentication user (`amldapuser`).

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

9. **After entering all the information, click Next.**

10. **Review the installation summary to ensure that the information you have 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 panel, 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 Apache HTTP Server instance on which you just installed the agent.**

Next Steps To ensure that the installation was successful, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 58.

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

Command-Line Installation of Agent for Apache HTTP Server on Solaris Systems

The following instructions describe how to use the command-line interface of the installation program to install a web agent.

To Install Agent for Apache HTTP Server on Solaris Systems Using the Command Line

Installing a web agent on a deployment container using the command line requires that you perform the following steps:

1. **Unpack the product binary in the directory of your choice using the following command:**

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

2. **In the directory in which you unpack the binaries, issue the following command:**

   ```
   # ./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 the web agent in this directory: Enter the full path to the directory in which you want to install the web agent.

   The directory you choose in which to install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

4. **Provide the following information about the Apache HTTP Server instance this agent will protect:**

   - Host Name
   - Apache Binary Directory
   - Web Server Port
   - Web Server Protocol
   - Agent Deployment URI
   - Apache Config Directory
   - SSL Ready

   For a description of the information to enter for these prompts, see "GUI Installation of Agent for Apache HTTP Server on Solaris Systems" on page 31.

5. **Provide the following information about the Access Manager host:**

   - 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-Access Manager Shared Secret
- Re-enter Shared Secret
- CDSSO Enabled

For a description of the information to enter for these prompts, see "GUI Installation of Agent for Apache HTTP Server on Solaris Systems" on page 31.

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 Java(tm) System Access Manager Policy Agent Installed   Available
2.  Done
```

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

8 Restart the Apache HTTP Server instance on which you just installed the agent.

Next Steps To ensure that the installation was successful, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 58

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

AIX Systems: Agent Installation for Apache HTTP Server

This section describes the installation process on AIX systems.

Preparing to Install Agent for Apache HTTP Server on AIX Systems

Follow the specific steps outlined in this section before you install the web agent to reduce the chance of complications occurring during and after the installation.

▼ To Prepare to Install Policy Agent 2.2 for Apache HTTP Server on AIX Systems

Note – 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 web agent installation program. Currently, JRE 1.3.1 or any version higher is certified for use with the web agent installation program.

Perform the following pre-installation tasks:

1. **Ensure that Policy Agent 2.2 for Apache HTTP Server is supported on the desired platform as listed in Table 2–1.**

2. **Install Apache HTTP Server if not already installed.**

   Refer to the Apache HTTP Server documentation for details on how best to install and configure this server for your platform.

3. **Ensure that Apache HTTP Server has the latest patches available.**

4. **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:
   ```

Installing Agent for Apache HTTP Server on AIX Systems

The web agent installation program for AIX systems has only a command-line interface. The instructions follow for installing this web agent:

Note – Unlike the behavior on other UNIX based platforms, no packages specific to AIX systems are installed by the agent installer. The installation process involves extracting the compressed files and executing a configuration script, which configures specified properties in the web agent `AMAgent.properties` configuration file.

Installation of Agent for Apache HTTP Server on AIX Systems

The following instructions describe how to use the command-line interface of the installation program to install Agent for Apache HTTP Server on AIX Systems.

▼ To Install Agent for Apache HTTP Server on AIX Systems Using the Command Line

Installing a web agent on a deployment container using the command line requires you to perform the following steps:

1. **Unpack the product binary in the directory of your choice using the following command:**

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

2. **Set `LIBPATH` to include the `libpasswd.so` file.**

   The `libpasswd.so` file is typically located in the directory in which the agent binaries are extracted. For example if `libpasswd.so` is in the directory `/export/apache_agent` , then `LIBPATH` should contain `/export/apache_agent`.

   In this case, using the Bash UNIX shell, you could set `LIBPATH` as follows:

   ```
   # LIBPATH=$LIBPATH:/export/apache_agent
   # export LIBPATH
   ```

3. **In the directory in which you unpack the binaries, issue the following command:**

   ```
   # ./setup -nodisplay
   ```

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 the web agent in this directory: Enter the full path to the directory in which you want to install the web agent.

The directory you choose in which to install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

5 Provide the following information about the Apache HTTP Server instance this agent will protect:

Host Name: Enter the fully qualified domain name (FQDN) of the machine where the Apache HTTP Server instance is installed.

For example, if the host is `host1`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host1.eng.example.com`.

Apache Binary Directory: Enter the full path to the directory where the Apache HTTP Server binary, therefore the `httpd` binary, is installed. An example pathname follows:

Apache-base`/bin`

where *Apache-base* represents the directory where Apache HTTP Server was installed. Refer to the Apache HTTP Server documentation for the specific path name.

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

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

Agent Deployment URI: Enter a Universal Resource Identifier (URI) that will be used to access Agent for Apache HTTP Server. The default value is `/amagent`.

Note – The web agent uses the value of the `com.sun.am.policy.agents.config.agenturi.prefix` property in the web agent `AMAgent.properties` configuration file to support some essential functions such as notification. 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 Apache HTTP Server instance where the agent is installed and *agent-deployment-uri* is the URI where the Apache HTTP Server instance will look for web-agent related HTML pages. Its default value is `amagent`.

The following is an example of an Agent Deployment URI:

```
http://host1.example.com:80/amagent
```

SSL Ready: Select this option if the Apache HTTP Server instance you are using has support for SSL. Your Apache HTTP Server instance 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 HTTP Server instance has been compiled with the EAPI flag, go to the `bin` directory of the Apache HTTP Server instance and type the following command:

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

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

The supported configurations for Apache HTTP Server are:

- Apache HTTP Server without `mod_ssl` support
- Apache HTTP Server with `mod_ssl` and EAPI flag enabled.

Note – Apache HTTP Server with `mod_ssl` support and EAPI flag disabled configuration is not supported by Policy Agent 2.2.

---

6 Provide the following information about the Access Manager host:

The web agent will connect to this server.

Primary Server Host: Enter the FQDN of the primary Access Manager host.

For example, if the host is `host3`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host3.eng.example.com`.

Primary Server Port: Enter the port number for the primary Access Manager host.

Primary Server Protocol: If the primary Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP.

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

Primary Console Deployment URI: Enter the location that was specified when Access Manager Console was installed. The default URI for Access Manager is `/amconsole`.

Failover Server Host: Enter the FQDN of the secondary Access Manager host if the primary Access Manager host becomes unavailable. If no failover server host exists, then leave this field blank.

Failover Server Port: Enter the port number of the secondary Access Manager host. If no failover server host exists, then leave this field blank.

Failover Server Protocol: If the failover Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP. If no failover server host exists, then leave this field blank.

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

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

Agent Access Manager Shared Secret: Enter the password for the Access Manager internal LDAP authentication user. This user is also referred to as `amldapuser`.

For more information about the shared secret and its relationship with the Access Manager agent profile, see Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2."

Re-enter Shared Secret: Re-enter the password for the Access Manager internal LDAP authentication user (`amldapuser`).

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

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 Java(tm) System Access Manager Policy Agent Installed   Available
2.  Done
```

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

9 **Restart the Apache HTTP Server instance on which you just installed the agent.**

Next Steps To ensure that the installation was successful, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 58.

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

Linux Systems: Agent Installation for Apache HTTP Server

This section describes the installation process on Linux systems.

Preparing to Install Agent for Apache HTTP Server on Linux Systems

Follow the tasks outlined in this section before you install the web agent. The first pre-installation task applies specifically to Linux systems. The second pre-installation task involves general steps that are not Linux specific.

▼ To Prepare to Install Agent for Apache HTTP Server Specifically on Linux Systems

If you are installing the agent for Apache HTTP Server on a Linux system, you must complete the following tasks in the order they are listed below, to ensure that Apache HTTP Server is configured with the POSIX Threads library. Failing to perform these steps might result in the application becoming unusable or might result in the entire system becoming unstable and unusable.

1 **Get the Apache HTTP Server source (version 1.3.33 or 2.0.54) from http://httpd.apache.org/ (`http://httpd.apache.org/`)**

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

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

3 **Configure your version of Apache HTTP Server with the respective flags as follows:**

- Apache HTTP Server 1.3.33

  Apache-source`/configure --prefix=`*Apache-base* `\`
  `--enable-rule=SHARED_CORE --enable-shared=max`

- Apache HTTP Server 2.0.54

Apache-source`/configure --prefix=`*Apache-base* `--enable-so`

Apache-source represents the directory where the Apache HTTP Server source was unpacked

Apache-base represents the directory where Apache HTTP Server was installed

4 **Rebuild and install Apache HTTP Server.**

Refer to the Apache HTTP Server documentation for details on how best to install and configure this server.

5 **Ensure that Apache HTTP Server has the latest patches available.**

▼ To Prepare to Install Agent for Apache HTTP Server on Linux Systems

Note – 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 web agent installation program. Currently, JRE 1.3.1 or any version higher is certified for use with the web agent installation program.

Perform the following pre-installation tasks:

1 **Ensure that Policy Agent 2.2 for Apache HTTP Server is supported on the desired platform as listed in "Supported Platforms and Compatibility of Agent for Apache HTTP Server" on page 25.**

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, 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:
```

Installing Agent for Apache HTTP Server on Linux Systems

The web agent installation program has two interfaces: the graphical user interface (GUI) and the command-line Interface. The following sections present instructions to install the web agent using both of these interfaces:

GUI Installation of Agent for Apache HTTP Server on Linux Systems

Use the following instructions to install the web agent using the GUI on the Linux systems.

▼ To Install Agent for Apache HTTP Server on Linux Systems Using the GUI

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

1 **Unpack the product binaries.**

Unpack the product binary in the directory of your choice using the following command:

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

2 **In the directory in which you unpack the binaries, issue the following command:**

```
# ./setup
```

The Welcome page appears.

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 web agent, click Browse. To accept the default, click Next.**

6 **When prompted, provide the following information about the Apache HTTP Server instance this agent will protect:**

Install Sun Java System Access Manager Policy Agent in this directory: Enter the full path to the directory where you want this agent to be installed, and then click Next.

The directory you choose in which to install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

Host Name: Enter the FQDN of the machine where the Apache HTTP Server instance is installed. For example, if the host is `host1`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host1.eng.example.com`.

Apache Binary Directory: Enter the full path to the directory where the Apache HTTP Server binary, therefore the `httpd` binary, is installed. An example pathname follows:

Apache-base`/bin`

where *Apache-base* represents the directory where Apache HTTP Server was installed. Refer to the Apache HTTP Server documentation for the specific path name.

Web Server Port: Enter the port number for the Apache HTTP Server instance that will be protected by the agent.

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

Agent Deployment URI: Enter a Universal Resource Identifier (URI) that will be used to access Agent for Apache HTTP Server. The default value is `/amagent`.

Note – The web agent uses the value of the `com.sun.am.policy.agents.config.agenturi.prefix` property in the web agent `AMAgent.properties` configuration file to support some essential functions such as notification. 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 Apache HTTP Server instance where the agent is installed and *agent-deployment-uri* is the URI where the Apache HTTP Server instance will look for web-agent related HTML pages. Its default value is `amagent`.

The following is an example of an Agent Deployment URI:

```
http://host1.example.com:80/amagent
```

Apache Config Directory: Enter the full path to the directory that contains the Apache HTTP Server configuration file `httpd.conf`. An example pathname follows:

Apache-base`/conf`

where *Apache-base* represents the directory where Apache HTTP Server was installed.

SSL Ready: Select this option if the Apache HTTP Server instance you are using has support for SSL. Your Apache HTTP Server instance 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 HTTP Server instance has been compiled with the EAPI flag, go to the `bin` directory of the Apache HTTP Server instance and type the following command:

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

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

The supported configurations for Apache HTTP Server are:

- Apache HTTP Server without `mod_ssl` support
- Apache HTTP Server with `mod_ssl` and EAPI flag enabled.

Note – Apache HTTP Server with `mod_ssl` support and EAPI flag disabled configuration is not supported by Policy Agent 2.2.

---

7 When you have entered all the information, click Next.

8 Enter information about the Access Manager host.

The web agent will connect to this server.

Primary Server Host: Enter the fully qualified domain name (FQDN) of the primary Access Manager host.

For example, if the host is `host3`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `host3.eng.example.com`.

Primary Server Port: Enter the port number for the primary Access Manager host.

Primary Server Protocol: If the primary Access Manager host is SSL-enabled, select HTTPS; otherwise select HTTP.

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

Primary Console Deployment URI: Enter the location that was specified when Access Manager Console was installed. The default URI for Access Manager is `/amconsole`.

Failover Server Host: Enter the FQDN of the secondary Access Manager host if the primary Access Manager host becomes unavailable. If no failover host exists, then leave this field blank.

Failover Server Port: Enter the port number of the secondary Access Manager host. If no failover host exists, then leave this field blank.

Failover Server Protocol: If the failover Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP. If no failover server host exists, then leave this field blank.

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

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

Agent Access Manager Shared Secret: Enter the password for the Access Manager internal LDAP authentication user. This user is also referred to as `amldapuser`.

For more information about the shared secret and its relationship with the Access Manager agent profile, see Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2."

Re-enter Shared Secret: Re-enter the password for the Access Manager internal LDAP authentication user (`amldapuser`).

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

9. **After entering all the information, click Next.**

10. **Review the installation summary to ensure 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 the Apache HTTP Server instance on which you just installed the agent.**

Next Steps To ensure that the installation was successful, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 58.

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

Command Line Installation of Agent for Apache HTTP Server on Linux Systems

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

▼ To Install Agent for Apache HTTP Server on Linux Systems Using the Command Line

Use the following instructions to install the web agent using the command line on Linux systems.

1. **Unpack the product binary in the directory of your choice using the following command:**

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

2. **In the directory in which you unpack the binaries, issue the following command:**

   ```
   # ./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 the agent in this directory: Enter the full path to the directory in which you want to install the agent.

The directory you choose in which to install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

4 Provide the following information about the Apache HTTP Server instance this agent will protect:

- Host Name
- Apache Binary Directory
- Web Server Port
- Web Server Protocol
- Agent Deployment URI
- Apache Config Directory
- SSL Ready

For a description of the information to enter for these prompts, see "GUI Installation of Agent for Apache HTTP Server on Linux Systems" on page 43.

5 Provide the following information about the Access Manager host:

- 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-Access Manager Shared Secret
- Re-enter Shared Secret
- CDSSO Enabled

  For a description of the information to enter for these prompts, see "GUI Installation of Agent for Apache HTTP Server on Linux Systems" on page 43.

  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 Java(tm) System Access Manager Policy Agent Installed   Available
2.  Done
```

7 **To see log information, enter** 1**. To exit the Installation program, enter** 2**.**

Next Steps To ensure that the installation was successful, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 58.

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

Windows Systems: Agent Installation for Apache HTTP Server

This section describes the installation process on Windows systems.

Preparing To Install Agent for Apache HTTP Server on Windows Systems

Follow the specific steps outlined in this section before you install the web agent to reduce the chance of complications occurring during and after the installation.

▼ To Prepare To Install Agent for Apache HTTP Server on Windows Systems

Note – 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 web agent installation program. Currently, JRE 1.3.1 or any version higher is certified for use with the web agent installation program.

Perform the following pre-installation tasks:

1. **Ensure that Policy Agent 2.2 for Apache HTTP Server is supported on the desired platform as listed in "Supported Platforms of Agent for Apache HTTP Server" on page 25.**

2. **Install Apache HTTP Server if not already installed.**

   Refer to the Apache HTTP Server documentation for details on how best to install and configure this server for your platform.

3. **Ensure that Apache HTTP Server has the latest patches available.**

4. **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:
   ```

Installing Agent for Apache HTTP Server on Windows Systems

The installation program that installs Agent for Apache HTTP Server has one interface, a graphical user interface (GUI).

The installation performed by this installer is extremely basic. The installer performs the following:

- Provides the license agreement
- Distributes the agent files

Therefore, during the installation, you are not prompted for information about the Apache HTTP Server host or the Access Manager host, though this type of information is often

prompted by installers. Instead, for this agent, such information is prompted as part of the configuration process described in "Windows Systems: Installation-Related Configuration for Apache HTTP Server" on page 52.

▼ To Install Agent for Apache HTTP Server on Windows Systems

You must have administrator privileges to run the installation program.

1. **Unzip the product binaries.**

   `unzip` *`binaryname`*`.zip`

   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. **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 it.**

5. **Select the directory in which you want to install the agent.**

   The default directory is C:`\Sun\Access_Manager\Agents\2.2.` The installation program will install the agent in this directory.

   The directory in which you install the web agent is referred to as the Policy Agent base directory, or *PolicyAgent-base*.

6. **(Conditional) Click Create Directory if this option is available.**

   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 you 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.

Windows Systems: Installation-Related Configuration for Apache HTTP Server

After you have performed the basic installation process, you must create a configuration file for the web site (or web sites) that is to be protected by the agent and then you must configure the agent for that web site (or web sites). These tasks are described in the following subsections:

- "Windows Systems: Creating Configuration Files, Agent for Apache HTTP Server" on page 52
- "Windows Systems: Configuring Agent for Apache HTTP Server for a Web Site" on page 56

Windows Systems: Creating Configuration Files, Agent for Apache HTTP Server

The agent for Apache HTTP Server provides a Visual Basic (VB) script to help you create agent configuration files. When you run it, the VB script prompts for information related to the Web Site Identifier, the agent you are installing, and Access Manager. The script creates an agent configuration file based on the information you provide.

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. Use the following steps to create multiple agent configuration files. However, ensure that you give a unique file name to each of the configuration files.

▼ Windows Systems: To Create Configuration Files, Agent for Apache HTTP Server

1 Change to the directory:

PolicyAgent-base`\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*

`ApacheCreateConfig.vbs`	is a VB script that saves your responses to prompts about the Apache HTTP Server host and the Access Manager host in a file. For this example, the file is represented by *defaultConfig*.
defaultConfig	represents the agent configuration file created by this command and for which you provide the actual name. This is a text file to which the output of the commands entered while running the script are written.

Note – Give a unique name for this agent 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. All the script prompts are displayed, for example purposes, in this step. However, information about the responses is presented in the subsequent steps.

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

Fully Qualified Host Name :
agentHost.com

Apache Binary Directory :
c:\program files\apache group\apache2\bin

Web Server Protocol [http] :


Web Server Port [80] :


Agent Deployment URI [/amagent] :

-------------------------------------------------
Sun Java (TM) Enterprise System Access Manager
-------------------------------------------------
Primary Server Host :
amHost.com

Primary Server Protocol [http] :


Primary Server Port Number [58080] :


Primary Server Deployment URI [/amserver] :
```

```
Primary Server Console URI [/amconsole] :


Failover Server Host :


Agent-Access Manager Shared Secret :

Re-enter Shared Secret :

CDSSO Enabled [false] :

-----------------------------------------------
Agent Configuration file created ==>  agentConfig
Execute the below command for Agent Configuration :
      cscript.exe ApacheAdmin.vbs -config agentConfig
-----------------------------------------------
```

3 When prompted, provide the following information about the Apache HTTP Server instance that this agent will protect:

Agent Resource File Name: Accept the default for this prompt (`ApacheResource.en`).

Host Name: Enter the fully qualified domain name (FQDN) of the system on which Apache HTTP Server is installed.

For example, if the host is `agentHost`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `agentHost.eng.example.com`.

Server Protocol: If this instance of Apache HTTP Server has been configured for SSL, then select HTTPS; otherwise select HTTP.

Server Port: Enter the port number of the Apache HTTP Server instance that will be protected by the agent.

Agent Deployment URI: Enter a Universal Resource Identifier (URI) that will be used to access Agent for Apache HTTP Server. The default value is `/amagent`.

Note – The web agent uses the value of the `com.sun.am.policy.agents.config.agenturi.prefix` property in the web agent `AMAgent.properties` configuration file to support some essential functions such as notification. 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 Apache HTTP Server instance where the agent is installed and *agent-deployment-uri* is the URI where the Apache HTTP Server instance will look for web-agent related HTML pages. Its default value is `amagent`.

The following is an example of an Agent Deployment URI:

```
http://agentHost.example.com:80/amagent
```

where the host name is `agentHost` and the domain name is `example.com`.

4 When prompted, provide the following information about the Access Manager host:

Primary Server Host: Enter the FQDN of the primary Access Manager host.

For example, if the host is `amHost`, the subdomain is `eng`, and the domain is `example.com`, then the Host Name in this case is `amHost.eng.example.com`.

Primary Server Protocol: If the primary Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP.

Primary Server Port: Enter the port number for the primary Access Manager host.

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

Primary Console Deployment URI: Enter the location that was specified when Access Manager Console was installed. The default URI for Access Manager is `/amconsole`.

Failover Server Host: Enter the FQDN of the secondary Access Manager host if the primary Access Manager host becomes unavailable. If no failover server host exists, then leave this field blank.

Failover Server Port: Enter the port number of the secondary Access Manager host. If no failover server host exists, then leave this field blank.

Failover Server Protocol: If the failover Access Manager host is SSL-enabled, select HTTPS. Otherwise select HTTP. If no failover server host exists, then leave this field blank.

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

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

Agent Access Manager Shared Secret: Enter the password for the Access Manager internal LDAP authentication user. This user is also referred to as `amldapuser`.

For more information about the shared secret and its relationship with the Access Manager agent profile, see Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2."

Re-enter Shared Secret: Re-enter the password for the Access Manager internal LDAP authentication user (`amldapuser`).

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

With the information you provide, the script creates the agent configuration file for you to use to configure this agent as described in the following section.

Windows Systems: Configuring Agent for Apache HTTP Server for a Web Site

Configure Agent for Apache HTTP Server 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 "Windows Systems: Creating Configuration Files, Agent for Apache HTTP Server" on page 52.

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

▼ Windows Systems: To Configure Agent for Apache HTTP Server for a Web Site

1 Change to the directory:

PolicyAgent-base`\apache\bin`

2 Run the following command:

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

`ApacheAdmin.vbs`	is a VB script that uses the output of the `ApacheCreateConfig.vbs` script. The output was saved to a configuration file, which for this example is represented by *defaultConfig*.
`-config`	is the option that allows the output to be used to configure the web site.
defaultConfig	represents the agent configuration file created previously as described in "Windows Systems: To Create Configuration Files, Agent for Apache HTTP Server" on page 52.

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.x. Re-start your server instance
```

3 **Restart the web site.**

4 **Try accessing the web site (`http://`*fqdn:port*`/index.html`).**

This link should take you to the Access Manager login page. After a successful authentication, if the policy is properly defined, you should be able to view the resource.

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

PolicyAgent-base`\debug\apache_`*port*

where *port* is the port number of Apache HTTP Server.

Note – If you want to configure the agent for multiple web sites, you must follow the preceding steps for each of the web sites.

Next Steps The last step of this task addresses verification of the agent installation. See the section that follows (All Systems: Verifying a Successful Installation on Policy Agent 2.2) for an expanded explanation on verifying the agent installation.

If you want to configure multiple instances of Apache HTTP Server, you must set up multiple Apache HTTP Server Virtual Hosts, as described in "All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts" on page 65.

All Systems: Verifying a Successful Installation on Policy Agent 2.2

After installing a web agent, ensure that the agent is installed successfully. Two methods are available for verifying a successful web agent installation. Perform both for best results.

▼ To Verify a Successful Installation

1 **Attempt to access a resource on the deployment container where the agent is installed.**

If the web agent is installed correctly, accessing any resource should take you to the Access Manager login page. After a successful authentication, if the policy is properly defined, you should be able to view the resource.

2 **Check the web agent `AMAgent.properties` configuration file.**

Make sure that each property is set properly. For information on the properties in this file, see Appendix C, "Web Agent `AMAgent.properties` Configuration File".

◆ ◆ ◆ CHAPTER 4

The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2

This section describes how to create or update an agent profile in Access Manager Console and then how to make the corresponding changes in the web agent.

If you are only interested in resetting the shared secret in the web agent, not the agent profile name, see "Resetting the Shared Secret Password" on page 94. However, first read the introductory paragraphs that follow in this section to become acquainted with the process and terminology related to the credentials used by web agents to authenticate with Access Manager. A common reason to reset only the shared secret is that it was entered incorrectly when prompted for during the installation of the web agent.

A web agent uses a user name and password as credentials to authenticate with Access Manager. You can use the default values for these credentials or you can create an agent profile in Access Manager Console and use those credentials. In web agents, the term for the default user name is agent user name. The default value of the agent user name is `UrlAccessAgent`. The term for the default password is shared secret. The default value of the shared secret is the password of the Access Manager internal LDAP authentication user. This user is commonly referred to as `amldapuser`.

Creating an agent profile is not a requirement for web agents. You can use the default values and never change the agent user name or shared secret. However, in certain situations you might want to change these default values. Changing the default values of the agent user name and shared secret involves creating an agent profile using Access Manager Console.

The terms used for the credentials are different once you create them in the agent profile. Agent user name is then called agent profile name. Shared secret is then called agent profile password. After you create the agent profile, you must assign the values of the agent profile name and the agent profile password to the correct properties in the web agent `AMAgent.properties` configuration file.

Creating or Updating a Web Agent Profile

The instructions that follow in this section explain how to change both the agent profile name and the agent profile password on the Access Manager side.

Since the agent profile is created and updated in Access Manager Console, tasks related to the agent profile are discussed in Access Manager documentation. Nonetheless, tasks related to the agent profile are also described in this Policy Agent guide, specifically in this chapter. For related information about defining the Policy Agent profile in Access Manager Console, see the following section of the respective document: "Agents" in *Sun Java System Access Manager 7 2005Q4 Administration Guide*.

▼ To Create or Update an Agent Profile in Access Manager

Perform the following tasks in Access Manager Console. The key steps of this task involve creating an agent ID (agent profile name) and an agent profile password.

1. **With the Access Control tab selected click the name of the realm for which you would like to create an agent profile.**
2. **Select the Subjects tab.**
3. **Select the Agent tab.**
4. **Click New.**
5. **Enter values for the following fields:**

   ID. Enter the agent profile name or identity of the agent.

   This is the agent profile name, which is the name the agent uses to log into Access Manager. Multi-byte names are not accepted. Do not use the web agent default value of `UrlAccessAgent`.

   Password. Enter the agent profile password.

   Do not use the web agent default value of this password. The web agent default value of this password is the password of the internal LDAP authentication user, commonly referred to as `amldapuser`.

   Password (confirm). Confirm the password.

   Device Status. Select the device status of the agent. The default status is Active. If set to Active, the agent will be able to authenticate to and communicate with Access Manager. If set to Inactive, the agent will not be able to authenticate to Access Manager.

6. **Click Create.**

   The list of agents appears.

7. **(Optional) If you desire, add a description to your newly created agent profile:**

   a. **Click the name of your newly created agent profile in the agent list.**

   b. **In the Description field, enter a brief description of the agent.**

      For example, you can enter the agent instance name or the name of the application it is protecting.

   c. **Click Save.**

Updating the Agent Profile Name and the Agent Profile Password in Web Agents

After you have changed the agent profile in Access Manager Console, assign the values for the agent profile name and the agent profile password to the corresponding properties in the web agent `AMAgent.properties` configuration file. This process involves the following:

- Adding the agent profile name to the following property in the web agent `AMAgent.properties` configuration file: `com.sun.am.policy.am.username`
- Encrypting the agent profile password (shared secret) using the encryption utility
- Adding the encrypted agent profile password (shared secret) to the following property in the web agent `AMAgent.properties` configuration file: `com.sun.am.policy.am.password`

The procedures specified in the preceding list are detailed in the platform-specific task descriptions that follow. Implement the steps according to the platform on which the web agent is installed.

▼ To Update the Agent Profile Name and Agent Profile Password on Solaris Systems

1. **Update the following property in the web agent `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username`

   Replace the value of this property with the agent profile name you just updated in Access Manager Console.

2. **Go to the following directory:**

   PolicyAgent-base`/SUNWam/agents/bin`

3. **Execute the following script in the command line:**

   `# ./crypt_util` *agent-profile-password*

   where *agent-profile-password* represents the agent profile password you just updated in Access Manager Console.

4. **Copy the output obtained after issuing the `# ./crypt_util` *agent-profile-password* command and paste it as the value for the following property:**

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

5. **Restart the deployment container and try accessing any resource protected by the agent.**

   If the agent gets redirected to Access Manager, this indicates the above steps were executed properly.

▼ To Update the Agent Profile Name and Agent Profile Password on AIX Systems

1. **Update the following property in the web agent `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username`

   Replace the value of this property with the agent profile name you just updated in Access Manager Console.

2. **Go to the following directory:**

   PolicyAgent-base`/agents/bin`

3. **Execute the following script in the command line:**

   `# ./crypt_util` *agent-profile-password*

   where *agent-profile-password* represents the agent profile password you just updated in Access Manager Console.

4. **Copy the output obtained after issuing the `# ./crypt_util` *agent-profile-password* command and paste it as the value for the following property:**

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

5. **Restart the deployment container and try accessing any resource protected by the agent.**

   If the agent gets redirected to Access Manager, this indicates the above steps were executed properly.

▼ To Update the Agent Profile Name and Agent Profile Password on Linux Systems

1. **Update the following property in the web agent `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username`

   Replace the value of this property with the agent profile name you just updated in Access Manager Console.

2. **Go to the following directory:**

   PolicyAgent-base`/bin`

3. **Execute the following script in the command line:**

   `crypt_util` *agent-profile-password*

   where *agent-profile-password* represents the agent profile password you just updated in Access Manager Console.

4. **Copy the output obtained after issuing the `crypt_util` *agent-profile-password* command and paste it as the value for the following property:**

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

5. **Restart the deployment container and try accessing any resource protected by the agent.**

   If the agent gets redirected to Access Manager, this indicates the above steps were executed properly.

▼ To Update the Agent Profile Name and Agent Profile Password on Windows Systems

1. **Update the following property in the web agent `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username`

   Replace the value of this property with the agent profile name you just updated in Access Manager Console.

2. **Go to the following directory:**

   PolicyAgent-base`\bin`

3. **Execute the following script from the command line**

   `cryptit` *agent-profile-password*

where *agent-profile-password* represents the agent profile password you just updated in Access Manager Console.

4 **Copy the output obtained after issuing the** `cryptit` *agent-profile-password* **command and paste it as the value for the following property:**

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

5 **Restart the deployment container and try accessing any resource protected by the agent.**

If the agent gets redirected to Access Manager, this indicates the above steps were executed properly.

◆ ◆ ◆ CHAPTER 5

Post-Installation Configuration: Policy Agent 2.2 for Apache HTTP Server

This chapter describes configuration and other post-installation considerations and tasks regarding Policy Agent 2.2 for Apache HTTP Server on the supported platforms as follows:

This chapter covers a few configuration tasks of Policy Agent 2.2 for Apache HTTP Server. The major tasks covered are the configuration of the agent on multiple virtual hosts of Apache HTTP Server and the configuration of SSL with the agent. Perform the tasks described in this chapter if they apply to your site's deployment.

After completing the applicable tasks described in this chapter, perform the tasks to configure the web agent to your site's specific needs as explained in Chapter 6, "Managing Policy Agent 2.2 for Apache HTTP Server."

All Systems: Configuring Agent for Apache HTTP Server on Multiple Apache HTTP Server Virtual Hosts

The task that follows applies to all platforms and provides an example of how to configure multiple virtual hosts. Therefore, if you are interested in configuring multiple instances of this agent, implement the task that follows.

▼ To Enable Access to Multiple Virtual Hosts

For this task example, two virtual hosts are configured: `http://site1.example.com` and `http://site2.example.com`. These host names are only examples.

1. **Define the FQDN map property in the web agent `AMAgent.properties` configuration file as follows:**

   ```
   com.sun.am.policy.agents.config.fqdn.map =
   valid1|site1.example.com,valid2|site2.example.com
   ```

2. **Define policies in Access Manager with virtual host names in the policy rules.**

Solaris Systems: Configuring Agent for Apache HTTP Server

This section provides task descriptions about using SSL for Solaris systems.

Solaris Systems: Using SSL With Agent for Apache HTTP Server

During installation, if you choose the HTTPS protocol, the Agent for Apache HTTP Server is automatically configured and ready to communicate over Secure Sockets Layer (SSL). Before proceeding with the tasks in this section, ensure that the Apache HTTP Server instance is configured for SSL.

Caution – 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 Apache HTTP Server.

Default Trust Behavior of Agent for Apache HTTP Server on Solaris Systems

This section only applies when Access Manager itself is running SSL. By default, the web agent installed on a remote Apache HTTP Server instance trusts any server certificate presented over SSL by the Access Manager host. The web agent does not check the root Certificate Authority (CA) certificate. If the Access Manager host is SSL-enabled and you want the web agent to perform certificate checking, adhere to the guidelines as described in the following subsections:

- "Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Solaris Systems" on page 66
- "Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on Solaris Systems" on page 67

Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Solaris Systems

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

`com.sun.am.trust_server_certs`

With this property set to `true`, the web agent does not perform certificate checking. On Solaris systems, setting this property to `false` is one of the steps involved in enabling the web agent to perform certificate checking as illustrated in the following task.

▼ To Disable the Default Trust Behavior of Agent for Apache HTTP Server on Solaris Systems

1. **Set the following property in the web agent `AMAgent.properties` configuration file to `false` as follows:**

   ```
   com.sun.am.trust_server_certs = false
   ```

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

   `com.sun.am.sslcert.dir =` *Apache-base*`/conf/cert`

   where *Apache-base* represents the directory where Apache HTTP Server was installed.

3. **(Conditional) Set the Cert DB Prefix.**

   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.certdb.prefix`

   Set the property in the following manner:

   `com.sun.am.certdb.prefix = https-`*host.domain.com.host*`-`

Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on Solaris Systems

The root CA certificate that you install on the remote instance of Apache HTTP Server must be the same one that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on Apache HTTP Server on Solaris Systems

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

1. **Change directories to the location of the Apache HTTP Server configuration file.**

   The following example is applicable for changing directories using the C shell when the Apache HTTP Server configuration file is in the default location of `/etc/apache/`:

   ```
   # /etc/apache/
   ```

2. **Change to the `cert` directory.**

3 Set the proper environment by issuing the following command:

```
# setenv LD_LIBRARY_PATH
PolicyAgent-base/SUNWam/agents/apache/lib:PolicyAgent-base/SUNWam/
agents/lib:/usr/lib/mps
```

4 (Conditional) If you have not already created the necessary certificate database, create that database now by issuing the following command:

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

5 Install root CA certificate by issuing the following command:

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

cert-name The name for this root CA certificate

cert-dir The directory where the certificate and key stores are located

cert-file The base-64 encoded root CA certificate file.

For more information on the `cetrutil` utility enter `cetrutil -H for Help.`

6 To verify that the certificate is properly installed, in the command line, issue the following command:

```
# PolicyAgent-base/SUNWam/agents/apache/cert/certutil -L -d .
```

The root CA certificate is then listed in the output of the `certutil -L` command as illustrated in the following code 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
```

7 Restart Apache HTTP Server.

AIX Systems: Configuring Agent for Apache HTTP Server

This section provides task descriptions for the following procedures:

- "AIX Systems: Setting File Ownership and Permissions on Agent for Apache HTTP Server" on page 69
- "AIX Systems: Using SSL With Agent for Apache HTTP Server" on page 70

After you check the file ownership and permissions (and reset if necessary), enable access to the proper libraries, and perform the procedure for verifying a successful installation. Next, determine if the remaining procedures described in this section apply to your site's deployment scenario. Perform the applicable procedures.

AIX Systems: Setting File Ownership and Permissions on Agent for Apache HTTP Server

On AIX systems, the Apache HTTP Server server must run as a non-root user. For example purposes in this section, the name `apuser` is used as the non-root user while `apgroup` is used as the name of the group.

To enable Agent for Apache HTTP Server to work properly, ensure that the non-root user has read permissions to the following files:

- `/etc/opt/agents/apache/config/`*_PathInstanceName*`/AMAgent.properties`
- `/var/opt/agents/apache/debug/`*_PathInstanceName*`/amAgent`
- *PolicyAgent-base*`/agents/apache/lib/`

PolicyAgent-base represents the directory you choose in which to install the web agent

_PathInstanceName represents a directory that is created and named during agent installation. This name is derived from the path to the Apache HTTP Server directory where slashes are converted to underscores. For this example, the path to the Apache HTTP Server directory is as follows:

```
/usr/local/apache2054
```

Based on the preceding path, during installation, the following *_PathInstanceName* directory would be created:

```
_user_local_apache2054_conf
```

You can set the required permissions to the files by issuing the following commands:

```
chown apuser:apgroup /etc/opt/agents/apache/config/_PathInstanceName
chown apuser:apgroup /var/opt/agents/apache/debug/_PathInstanceName/
```

```
chown apuser:apgroup PolicyAgent-base/agents/apache/lib/libamapc2.a
```

AIX Systems: Using SSL With Agent for Apache HTTP Server

During installation, if you choose the HTTPS protocol, Agent for Apache HTTP Server is automatically configured and ready to communicate over Secure Sockets Layer (SSL). Before proceeding with tasks in this section, ensure that the Apache HTTP Server instance is configured for SSL.

Caution – 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 Apache HTTP Server.

Default Trust Behavior of Agent for Apache HTTP Server on AIX Systems

This section only applies when Access Manager itself is running SSL. By default, the web agent installed on a remote Apache HTTP Server instance trusts any server certificate presented over SSL by the Access Manager host. The web agent does not check the root Certificate Authority (CA) certificate. If the Access Manager host is SSL-enabled and you want the agent to perform certificate checking, adhere to the guidelines as described in the following subsections:

- "Disabling the Default Trust Behavior of Agent for Apache HTTP Server on AIX Systems" on page 70
- "Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on AIX Systems" on page 71

Disabling the Default Trust Behavior of Agent for Apache HTTP Server on AIX Systems

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

```
com.sun.am.trust_server_certs
```

With this property set to true, the web agent does not perform certificate checking. On AIX systems, setting this property to false is one of the steps involved in enabling the web agent to perform certificate checking as illustrated in the following task.

▼ To Disable the Default Trust Behavior of Agent for Apache HTTP Server on AIX Systems

1 **Set the following property in the web agent** `AMAgent.properties` **configuration file to** `false` **as follows:**

```
com.sun.am.trust_server_certs = false
```

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

```
com.sun.am.sslcert.dir = Apache-base/conf/cert
```

where *Apache-base* represents the directory where Apache HTTP Server was installed.

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.certdb.prefix
```

Set the property as follows:

```
com.sun.am.certdb.prefix = https-host.domain.com.host-
```

Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on AIX Systems

The root CA certificate that you install on the remote instance of Apache HTTP Server must be the same certificate that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on Apache HTTP Server on AIX Systems

The following steps outline a method for installing Access Manager Root CA Certificate on the Apache HTTP Server server. However, see the documentation for the Apache HTTP Server server for more information about installing certificates.

1 **Change directories to the location of the Apache HTTP Server configuration file.**

The following example is applicable for changing directories using the C shell when the Apache HTTP Server configuration file is in the default location of *Apache-base* `/conf/`:

```
# Apache-base/conf/
```

where *Apache-base* represents the directory where Apache HTTP Server was installed.

2 Change to the `cert` directory.

This `cert` directory is created by the agent installer.

3 Set the proper environment by issuing a command such as the following (using the tcsh shell, for example):

```
# setenv LIBPATH
PolicyAgent-base/agents/apache/lib:$LIBPATH
```

4 (Conditional) If you have not already created the necessary certificate database, create that database now by issuing the following command:

```
# PolicyAgent-base/agents/bin/certutil -N -d .
```

5 Install root CA certificate by issuing the following command:

```
# PolicyAgent-base/agents/bin/certutil -A -n cert-name -t
"C,C,C" -d cert-dir -i cert-file
```

cert-name The name for this root CA certificate

cert-dir The directory where the certificate and key stores are located

cert-file The base-64 encoded root CA certificate file.

For example, if the Root CA certificate of the Access Manager host is present in the current directory, which is *Apache-base*`/conf/cert`, and if the name of this certificate file is `root_ca.crt`, then execute the following command:

```
/usr/local/apg/agents/bin/certutil -A -n am_root_ca_cert -t "C,C,C" -d . -i root_ca.crt
```

For this example, *PolicyAgent-base* is `/usr/local/apg`.

For more information on the `cetrutil` utility enter `cetrutil -H for Help.`

6 To verify that the certificate is properly installed, in the command line, issue the following command:

```
# PolicyAgent-base/agents/bin/certutil -L -d .
```

The root CA certificate is then listed in the output of the `certutil -L` command as illustrated in the following code 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
```

7 **Restart Apache HTTP Server.**

Linux Systems: Configuring Agent for Apache HTTP Server

This section provides task descriptions for Linux systems. Notice that the first task is specific to SUSE Linux. The tasks in this section are as follows:

- "Agent for Apache HTTP Server on SUSE Linux: Obtaining the Required Libraries" on page 73
- "Linux Systems: Using SSL With Agent for Apache HTTP Server" on page 74

Only perform a task if a respective condition applies.

Agent for Apache HTTP Server on SUSE Linux: Obtaining the Required Libraries

Agent for Apache HTTP Server supports SUSE Linux Enterprise 9 as described in Table 2–1. However, SUSE Linux Enterprise 9 does not contain certain shared libraries that are required by Agent for Apache HTTP Server. The following task describes how to make the required libraries available.

▼ To Obtain the Libraries Required by SUSE Linux

- **Issue the following command:**

  ```
  rpm --prefix=PolicyAgent-base -i common-2.2-0.i686.rpm
  ```

  This command installs the package `common-2.2`, which in turn installs the shared libraries from the agent binary.

Note – If you install this agent again, you must perform this task again to make the required libraries available. Also, if you uninstall this agent you must remove the `common-2.2` package as described in "Agent for Apache HTTP Server on SUSE Linux: Removing the `common-2.2` Package" on page 104.

Linux Systems: Using SSL With Agent for Apache HTTP Server

During installation, if you chose the HTTPS protocol, the Agent for Apache HTTP Serveris automatically configured and ready to communicate over Secure Sockets Layer (SSL). Before proceeding with the tasks in this section, ensure that the Apache HTTP Server instance is configured for SSL.

Caution – 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 Apache HTTP Server.

Default Trust Behavior of Agent for Apache HTTP Server on Linux Systems

This section only applies when Access Manager itself is running SSL. By default, the web agent installed on a remote Apache HTTP Server instance will trust any server certificate presented over SSL by the Access Manager host. The web agent does not check the root Certificate Authority (CA) certificate. If the Access Manager host is SSL-enabled and you want the agent to perform certificate checking, adhere to the guidelines as described in the following subsections:

- "Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Linux Systems" on page 74
- "Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on Linux Systems" on page 75

Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Linux Systems

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

```
com.sun.am.trust_server_certs
```

With this property set to true, the web agent does not perform certificate checking. On Linux systems, enabling the web agent to perform certificate checking is a one-step process that only involves setting this property to `false` as illustrated in the following task.

▼ To Disable the Default Trust Behavior of Agent for Apache HTTP Server on Linux Systems

- **Set the following property in the web agent `AMAgent.properties` configuration file to `false` as follows:**

  ```
  com.sun.am.trust_server_certs = false
  ```

Installing the Access Manager Root CA Certificate for a Remote Apache HTTP Server Instance on Linux Systems

The root CA certificate that you install on the remote instance of Apache HTTP Server must be the same one that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on Apache HTTP Server on Linux Systems

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

1. **Change directories to the location of the Apache HTTP Server configuration file.**

   The following example is applicable for changing directories using the C shell when the Apache HTTP Server configuration file is in the default location of `/etc/apache/`:

   ```
   # /etc/apache/
   ```

2. **Change to the `cert` directory.**

3. **Set the proper environment by issuing the following command:**

   ```
   # setenv LD_LIBRARY_PATH
   PolicyAgent-base/agents/apache/lib:PolicyAgent-base/agents/lib:/usr/lib/mps
   ```

4. **(Conditional) If you have not already created the necessary certificate database, create that database now by issuing the following command:**

   ```
   # PolicyAgent-base/agents/apache/cert/certutil -N -d .
   ```

5. **Install root CA certificate by issuing the following command:**

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

   cert-name The name for this root CA certificate

cert-dir The directory where the certificate and key stores are located

cert-file The base-64 encoded root CA certificate file.

For more information on the `cetrutil` utility enter `cetrutil -H for Help.`

6 To verify that the certificate is properly installed, in the command line, issue the following command:

```
# PolicyAgent-base/agents/apache/cert/certutil -L -d .
```

The root CA certificate is then listed in the output of the `certutil -L` command as illustrated in the following code 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
```

7 Restart Apache HTTP Server.

Windows Systems: Configuring Agent for Apache HTTP Server

This section provides task descriptions about using SSL for Windows systems.

Windows Systems: Using SSL With Agent for Apache HTTP Server

During installation, if you choose the HTTPS protocol, Agent for Apache HTTP Server is automatically configured and ready to communicate over SSL. Before proceeding with the tasks in this section, ensure that the Apache HTTP Server instance is configured for SSL.

Caution – 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 the Apache HTTP Server server.

Default Trust Behavior of Agent for Apache HTTP Server on Windows Systems

This section only applies when Access Manager itself is running SSL. By default, Agent for Apache HTTP Server trusts any server certificate presented over SSL by the Access Manager host. The web agent does not check the root Certificate Authority (CA) certificate. If the Access Manager host is SSL-enabled and you want the agent to perform certificate checking, adhere to the guidelines as described in the following subsections:

- "Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Windows Systems" on page 77
- "Installing the Access Manager Root CA Certificate on Apache HTTP Server on Windows Systems" on page 77

Disabling the Default Trust Behavior of Agent for Apache HTTP Server on Windows Systems

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

```
com.sun.am.trust_server_certs
```

With this property set to true, the web agent does not perform certificate checking. On Windows systems, enabling the web agent to perform certificate checking is a one-step process that only involves setting this property to `false` as illustrated in the following task.

▼ To Disable the Default Trust Behavior of Agent for Apache HTTP Server on Windows Systems

- **Set the following property in the web agent `AMAgent.properties` configuration file to `false` as follows:**

```
com.sun.am.trust_server_certs = false
```

Installing the Access Manager Root CA Certificate on Apache HTTP Server on Windows Systems

The root CA certificate that you install on the Apache HTTP Server instance that the agent protects must be the same certificate that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on Apache HTTP Server on Windows Systems

The following steps outline a method for installing Access Manager Root CA Certificate on the Apache HTTP Server server. However, see the documentation for the Apache HTTP Server server for more information about installing certificates.

1 Change directories to the location of the Apache HTTP Server configuration file.

The following example is applicable for changing directories using the CMD shell when the Apache HTTP Server configuration file is in the default location of `c:\program files\apache group\apache2`:

```
# c:\program files\apache group\apache2
```

2 Change to the `cert` directory.

3 (Conditional) If the certificate database has not yet been created, create it at a unique location using a command such as the following:

```
# PolicyAgent-base\bin\certutil -N -d .
```

4 Install the root CA certificate.

Remember that the root CA certificate that you install on the Apache HTTP Server server must be the same certificate that is installed on the Access Manager host.

The following example demonstrates a command you can issue that uses the `certutil` utility to install the certificate:

```
# PolicyAgent-base\bin\certutil -A -n cert-name -t
"C,C,C" -d cert-dir -i cert-file
```

cert-name The name for this root CA certificate.

cert-dir The directory where the certificate and key stores are located.

cert-file The base-64 encoded root CA certificate file.

For more information on the `cetrutil` utility enter `cetrutil -H for Help.`

5 To verify that the certificate is properly installed, in the command line, issue the following command:

PolicyAgent-base`\bin\certutil -L -d` *cert-dir*

The root CA certificate is then listed in the output of the `certutil -L` command as illustrated in the following code 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
```

6 Restart Apache HTTP Server.

◆ ◆ ◆ CHAPTER 6

Managing Policy Agent 2.2 for Apache HTTP Server

Interaction with Policy Agent 2.2 for Apache HTTP Server is enabled through a limited number of scripts, such as an installation script, and by editing the web agent `AMAgent.properties` configuration file. This chapter describes how to modify the web agent accordingly.

This chapter focuses on methods available for managing this web agent, specifying the features you can configure and the tasks you can perform using each method as follows:

The section on tasks performed with the web agent `AMAgent.properties` configuration file provides details of how to perform these tasks while the section on tasks performed with web agent scripts simply summarizes the types of tasks you can perform with scripts.

Key Features and Tasks Performed with the Web Agent `AMAgent.properties` Configuration File

The web agent `AMAgent.properties` configuration file is a text file of configuration properties that you can modify to change web agent behavior. However, the content of this file is very sensitive. Changes made can result in changes in how the agent works. Errors made can cause the agent to malfunction.

This section focuses on some of the key features and tasks performed with the configuration file. For a list and description of every property in the configuration file, access the configuration file itself located as described in Table 6–1. Also a list of the properties is available in this guide, at Appendix C, "Web Agent `AMAgent.properties` Configuration File".

This section describes the most important details of the configuration file, such as how specific properties can be modified to produce specific results. The topics described are typically those

of greatest interest in real-world deployment scenarios. For a list and description of every property in the configuration file, access the configuration file itself located as described in Table 6–1. Also a list of the properties is available in this guide, at Appendix C, "Web Agent AMAgent.properties Configuration File".

This section describes the following:

Locating the Web Agent AMAgent.properties Configuration File

The following table provides the default location for the web agent AMAgent.properties configuration file on the supported platforms.

TABLE 6–1 Location of the Web Agent AMAgent.properties Configuration File

Server	Platform	Location
	Solaris (SPARC and X86)	/etc/opt/SUNWam/agents/apache/config/*_PathInstanceName/*
	AIX and Linux	/etc/opt/agents/apache/config/*_PathInstanceName/*
	Windows	*PolicyAgent-base*\apache\config\apache_*port*\

where *port* is the port number of the Apache HTTP Server instance and *_PathInstanceName* represents the name of a directory that is automatically created and named during the agent installation process. The name for this directory is derived from the full path name of the

Apache conf directory. The process of creating the directory name involves the conversion of forward slash symbols "/" into underscore symbols "_." For example, a full path name of /opt/apache/conf provides the information used to name this new directory as follows:

```
_opt_apache_conf
```

Using the Web Agent AMAgent.properties Configuration File

Changing the web agent AMAgent.properties configuration file can have serious and far-reaching effects. When you make 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 backslash (\).

Note – If you make changes to the web agent AMAgent.properties configuration file, restart the deployment container to make your changes take effect.

The web agent AMAgent.properties configuration file includes information for a variety of configurations, including the following:

- debugging
- fully qualified domain name (FQDN) map
- Access Manager services
- service and agent deployment descriptors
- session failover

The configuration file also contains configuration information on advanced features, such as forwarding LDAP user attributes through HTTP headers and POST data preservation.

Providing Failover Protection for a Web Agent

When you install a web agent, you can specify a *failover* or backup deployment container, such as a web server, for running Access Manager. This is essentially a high availability option. It ensures that if the deployment container that runs Access Manager service becomes unavailable, the web agent still processes access requests through a secondary, or failover, deployment container running Access Manager service.

Setting up failover protection for the web agent, requires modifying the web agent AMAgent.properties configuration file. However, you must first install two different instances of Access Manager on two separate deployment containers.

Then follow the instructions in this guide to about installing the web agent. The web agent installation program prompts you for the host name and port number of the failover deployment container that you have configured to work with Access Manager. The following property in the web agent AMAgent.properties configuration file, stores the failover deployment container name:

```
com.sun.am.policy.am.login.url
```

Set this property in order to store failover deployment container information. Given the values in the following list, the property would be set as shown in Example 6–1.

host1	Name of the primary Access Manager host.
host2	Name of the first failover Access Manager host.
host3	Name of the second failover Access Manager host.
example	Name of the domain.
58080	Default port number

EXAMPLE 6–1 Configuration Property Setting for Failover Protection of a Web Agent

```
com.sun.am.policy.am.login.url = http://host1.example.com:58080/
amserver/UI/Login http://host2.example.com:58080/amserver/UI/Login
http://host3.example.com:58080/amserver/UI/Login
```

A failover server name is configurable after it has been set during installation. When configuring this property, note that a space is required between each Access Manager login URL.

Changing the Web Agent Caching Behavior

Each web agent maintains a cache that stores the policies for every user's session. The cache can be updated by a cache polling mechanism.

Cache Updates

A web agent maintains a cache of all active sessions involving content that the agent protects. Once an entry is added to an agent's 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.polling.interval in the web agent AMAgent.properties configuration file determines the number of minutes an entry will remain in the web 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.

In a normal deployment situation, policy changes on the server are frequent, which requires sites to accept a certain amount of latency for web 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 com.sun.am.policy.am.polling.interval property, set it to the lower of the two:

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

Note – Sun Java System Policy Agent 2.2 for Apache HTTP Server does not support notifications. Therefore, updating the cache through a notification mechanism is not an available feature. However, since the notification mechanism is available for other agents in the Policy Agent 2.2 software set, a property exists in the web agent AMAgent.properties configuration file to control this feature. The property that controls the notification mechanism, com.sun.am.notification.enable, is set to false for this agent. Do not set this property to true for this agent as it might result in unexpected behavior.

Configuring the 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 web agent denies access to all resources on the deployment container that it protects. However, various resources (such as a web site or an application) available through a deployment container 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 and the cascading style sheets (CSS) that apply to these home pages. The user should be able to browse such pages without authenticating. For the home page example, all these resources need to be on the not-enforced URL list or the page will not be displayed properly. The property com.sun.am.policy.agents.config.notenforced_list is 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, or "inverted", scenario when all the resources on the deployment container, except a list of URLs, are open to any user. In that case, the property com.sun.am.policy.agents.config.notenforced_list.invert would be used to reverse the meaning of com.sun.am.policy.agents.config.notenforced_list. If it is set to true (by default it is set to false), then the not-enforced URL list would become the enforced list.

EXAMPLE 6–2 Configuration Property Settings for Not-Enforced URL List

The following are examples:

Scenario 1: Not-Enforced URL List

EXAMPLE 6–2 Configuration Property Settings for Not-Enforced URL List *(Continued)*

```
com.sun.am.policy.agents.config.notenforced_list.invert = false

com.sun.am.policy.agents.config.notenforced_list =
http://host1.example.com:80/welcome.html
http://host1.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 web agent.

Scenario 2: Inverted Not-Enforced URL List

```
com.sun.am.policy.agents.config.notenforced_list.invert = true

com.sun.am.policy.agents.config.notenforced_list =
 http://host1.example.com:80/welcome.html
 http://host1.example.com:80/banner.html
```

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

Caution – If feasible, keep this property set to `false` as such:

```
com.sun.am.policy.agents.config.notenforced_list.invert = false
```

A value of `false` reduces the chance of unintentionally allowing access to resources.

Configuring the Not-Enforced IP Address List

The `com.sun.am.policy.agents.config.notenforced_client_ip_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 web 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.config.do_sso_only` is used to specify if only authentication is enforced for URLs protected by the web agent. If this property is set to `true`

(by default it is set to false), it indicates that the web agent enforces authentication only, without enforcing policies. After a user logs onto Access Manager successfully, the web agent will not check for policies related to the user and the accessed URLs.

Providing Personalization Capabilities

Web agents in Policy Agent 2.2 can personalize page content for users in three distinct ways as described in the following subsections:

- "Providing Personalization With Session Attributes" on page 87
- "Providing Personalization With Policy-Based Response Attributes" on page 88
- "Providing Personalization With User Profile Attributes Globally" on page 89

Providing Personalization With Session Attributes

Web agents in Policy Agent 2.2 support a feature where a user's session attributes are fetched and set as headers or cookies. The following property responsible for this task:

```
com.sun.am.policy.agents.config.session.attribute.fetch.mode
```

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

- NONE
- HTTP_HEADER
- HTTP_COOKIE

When set to NONE, no session attributes are fetched and the com.sun.am.policy.agents.config.session.attribute.map property is ignored. With this property set to either HTTP_HEADER or HTTP_COOKIE, the web agent fetches session attributes. Use the following property to configure attributes that are to be forwarded as HTTP headers or cookies: com.sun.am.policy.agents.config.session.attribute.map.

The following content is from the web agent AMAgent.properties configuration file. The text has been reformatted for this section. This section illustrates how the com.sun.am.policy.agents.config.session.attribute.map property maps session attributes to headers or cookies.

Session attributes are added to an HTTP header following this format:

```
session_attribute_name|http_header_name[,...]
```

The value of the attribute being fetched in session is session_attribute_name. This value gets mapped to a header value as follows: http_header_name.

Note – In most cases, in a destination application where http_header_name appears as a request header, it is prefixed with HTTP_ and the following type of conversion takes place:

Lower case letters	convert to upper case letters.
Hyphen "-"	converts to underscore "_"
"common-name"	as an example, converts to "HTTP_COMMON_NAME."

```
com.sun.am.policy.agents.config.session.attribute.map =
successURL | success-url, contextId | context-id
```

The session attribute is forwarded as a header or a cookie as determined by the end-user applications on the web container that the web agent is protecting. These applications can be considered the consumers of the forwarded header values. The forwarded information is used for the customization and personalization of web pages. You can also write server side plug-ins to put any user session attribute and define the corresponding attribute name and mapping in the preceding property to retrieve the value.

Providing Personalization With Policy-Based Response Attributes

Header attributes can also be determined by Access Manager policy configurations. With policy-based response attributes you can define attribute-value pairs at each policy.

Web agents in this release set policy-based response attributes as headers or cookies based on configuration. All subjects that match this attribute set obtain this attribute.

The following is a new property that has been added to the web agent AMAgent.properties configuration file to control this functionality:

```
com.sun.am.policy.agents.config.response.attribute.fetch.mode
```

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

- NONE
- HTTP_HEADER
- HTTP_COOKIE

The following example shows this configuration property with the default setting, which is HTTP_HEADER:

```
com.sun.am.policy.agents.config.response.attribute.fetch.mode = HTTP_HEADER
```

Attribute mapping is available for response attributes. Therefore, the format of policy information can be mapped to the format of a header or a cookie. The below property is used for this type of mapping:

```
com.sun.am.policy.agents.config.response.attribute.map
```

Unlike profile attributes and session attributes, where only the mapped attributes are displayed as headers or cookies, by default, response attributes are set by the agent as headers or cookies based on the setting of this property:

```
com.sun.am.policy.agents.config.response.attribute.fetch.mode
```

If a response attribute map is specified, then the corresponding attribute mapped name is fetched from the map and its corresponding value is displayed as either a header or a cookie based on the setting of the above property.

Providing Personalization With User Profile Attributes Globally

Web agents in Policy Agent 2.2 have the ability to forward user profile attribute values via HTTP headers to end-web applications. The user profile attribute values come from the server side of Access Manager. The web 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 web agent `AMAgent.properties` configuration file. To turn this feature on and off, edit the following property in the web agent `AMAgent.properties` configuration file:

```
com.sun.am.policy.agents.config.profile.attribute.fetch.mode
```

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

- `NONE`
- `HTTP_HEADER`
- `HTTP_COOKIE`

When set to `NONE`, the web agent does not fetch LDAP attributes from the server and ignores the `com.sun.am.policy.agents.config.profile.attribute.map` property. In the other two cases, the web agent fetches the attribute.

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

```
com.sun.am.policy.agents.config.profile.attribute.map
```

Below is an example section from the web agent `AMAgent.properties` configuration 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.agents.config.profile.attribute.map = 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 Access Manager is installed:

AccessManager-base/SUNWam/config/xml/amUser.xml

The attributes in this file could be either Access Manager user attributes or Access Manager dynamic attributes. For an explanation of these two types of user attributes, see *Sun Java System Access Manager 7 2005Q4 Administration Guide*.

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

Setting the Fully Qualified Domain Name

To ensure appropriate user experience, it is necessary that the users access resources protected by the web agent using valid URLs. The configuration property com.sun.am.policy.agents.config.fqdn.default provides the necessary information needed by the web agent to identify if the user is using a valid URL to access the protected resource. If the web 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 web 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 deployment container may not start up correctly. This property is set during the web agent installation and must not be modified unless absolutely necessary to accommodate deployment requirements. An invalid value for this property can result in the deployment container becoming unusable or the resources becoming inaccessible.

The property com.sun.am.policy.agents.config.fqdn.map provides another way by which the web agent can resolve partial or malformed access URLs and take corrective action. The web agent gives precedence to the entries defined in this property over the value defined in the com.sun.am.policy.agents.config.fqdn.default 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.config.fqdn.default property.

The com.sun.am.policy.agents.config.fqdn.map property can be used for creating a mapping for more than one hostname. This may be the case when the deployment container 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 deployment container resources becoming inaccessible.

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

```
com.sun.am.policy.agents.config.fqdn.map =
[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.config.fqdn.map = 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 web agent uses the name specified in this map instead of the deployment container's actual name.

If 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 in the Access Manager Console. In this file, set the mapping as com.sun.am.policy.agents.config.fqdn.map = valid|*xyz.hostname*.com.

Resetting Cookies

The cookie reset feature enables the web agent to reset some cookies in the browser session while redirecting to Access Manager for authentication.

This feature is configurable through two properties in the web agent AMAgent.properties configuration file.

- Enable Cookie Reset

  com.sun.am.policy.agents.config.cookie.reset.enable = true

  This property must be set to true if this web agent needs to reset cookies in the response while redirecting to Access Manager 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 Access Manager 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.config.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 web agent AMAgent.properties configuration file. To turn this feature on or off, use the following property:

```
com.sun.am.policy.agents.config.cdsso.enable = 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.config.cdcservlet.url
```

The following is an example of how this property could be set:

```
com.sun.am.policy.agents.config.cdcservlet.url =
http://host1.eng.example.com:58080/amserver/cdcservlet
```

The third property, `com.sun.am.policy.agents.config.cookie.domain.list` 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 web 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 CDSSO, see *Sun Java System Access Manager 7 2005Q4 Technical Overview*

Setting the REMOTE_USER Server Variable

The property `com.sun.am.policy.am.userid.param` allows you to configure the user ID parameter passed by the session or user profile information from Access Manager. The user ID value is used by the agent to set the value of the `REMOTE_USER` server variable. By default, this parameter is set to `UserToken` and is fetched from session attributes.

It can be set to any other session attribute. Another property determines where to retrieve the value, from user profiles or from session properties.

Example 1: This example demonstrates how to set the user ID parameter with session attributes:

`com.sun.am.policy.am.userid.param.type=SESSION` (this is default)

`com.sun.am.policy.am.userid.param=UserToken` (UserId, Principal, or any other session attribute)

Example 2: This example demonstrates how to set the user ID parameter with LDAP user profile attributes:

`com.sun.am.policy.am.userid.param.type=LDAP`

`com.sun.am.policy.am.userid.param=cn` (any profile attribute)

Setting Anonymous User

For resources on the not-enforced list, the default configuration does not allow the REMOTE_USER variable to be set. To enable the REMOTE_USER variable to be set for not-enforced URLs, you must set the following property in the web agent AMAgent.properties configuration file to TRUE (by default the value is FALSE):

```
com.sun.am.policy.agents.config.anonymous_user.enable = TRUE
```

When you set the value of this property to TRUE, the value of REMOTE_USER will be set to the value contained in the following property in the web agent AMAgent.properties configuration file:

```
com.sun.am.policy.agents.config.anonymous_user
```

By default, the value of this property is set to anonymous as follows:

```
com.sun.am.policy.agents.config.anonymous_user = anonymous
```

Validating Client IP Addresses

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

The web agent AMAgent.properties configuration file contains a property titled com.sun.am.policy.agents.config.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 the 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 balancer somewhere between the client browser and the agent-protected deployment container. In such cases, the IP address appearing in the request will not reflect the real IP address on which the client browser runs.

Resetting the Shared Secret Password

This section describes how to reset the shared secret. The web agent stores the shared secret in the web agent AMAgent.properties configuration file.

If you are only interested in resetting the shared secret, not the agent profile name, continue reading this section. If you are interested in creating or updating the agent profile in Access Manager Console and then updating the same credential information in the web agent, see Chapter 4, "The Relationship Between the Agent Profile and Web Agents in Policy Agent 2.2." The steps described in that chapter are comprehensive, integrating the simpler steps described in this section.

The chapter mentioned in the preceding paragraph also provides a useful explanation of the process and terminology related to the credentials used by web agents to authenticate with Access Manager. Refer to that chapter for more information.

This section specifically describes how to change the shared secret in web agents. The following situations might require you to reset the shared secret:

- You entered the shared secret incorrectly during web agent installation.
- You have been using the default shared secret, which is the `amldapuser` password, but this password has since been changed.

The value for the property `com.sun.am.policy.am.password` in the web agent `AMAgent.properties` configuration file is set with the encrypted shared secret during web agent installation. Therefore, if the shared secret is entered incorrectly during installation, the preceding property is assigned an incorrect value, preventing the web agent from authenticating with Access Manager.

To reset or change the shared secret, use the encryption utility to encrypt the shared secret and then set the value in the property as described in the following task.

▼ To Reset the Shared Secret

This task applies to all platforms.

1. **Go to the following directory:**

   PolicyAgent-base`/bin`

2. **Execute the appropriate platform-specific script in the command line:**

   - **UNIX-based Systems:**

     `# ./crypt_util` *shared-secret*

   - **Windows Systems:**

     `cryptit` *shared-secret*

where *shared-secret* represents the password, that along with the agent user name, allows the web agent to authenticate with Access Manager. The default value of the shared secret is the password of the Access Manager internal LDAP authentication user. This user is commonly referred to as amldapuser.

3 **Copy the output obtained after issuing the preceding command and paste it as the value for the following property:**

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

4 **Restart the deployment container and try accessing any resource protected by the agent.**

If the agent gets redirected to Access Manager, this indicates the above steps were executed properly.

Enabling Load Balancing

Various properties in the web agent AMAgent.properties configuration file can be used to enable load balancing. Edit the properties that apply, according to the location of the load balancer or load balancers in your deployment, as follows:

- "Load Balancer in Front of Access Manager" on page 96
- "Load Balancer in Front of Web Agent" on page 97
- "Load Balancers in Front of Both the Web Agent and Access Manager" on page 98

Load Balancer in Front of Access Manager

When a load balancer is deployed in front of Access Manager and a web agent interacts with the load balancer, the following properties must be edited:

```
com.sun.am.naming.url
com.sun.am.policy.am.login.url
com.sun.am.load_balancer.enable
```

EXAMPLE 6–3 Property Settings: Load Balancer in Front of Access Manager

This example illustrates property settings in the web agent AMAgent.properties configuration file that can be used to enable load balancing:

```
com.sun.am.naming.url = LB-url/amserver/namingservice
com.sun.am.policy.am.login.url = LB-url/amserver/UI/Login
com.sun.am.load_balancer.enable = true
```

where *LB-url* represents the load balancer URL. The following example is a conceivable load balancer URL:

EXAMPLE 6–3 Property Settings: Load Balancer in Front of Access Manager *(Continued)*

```
http://hostname.example.com:8080
```

Load Balancer in Front of Web Agent

In many cases, when a load balancer is deployed in front of the web agent only the following property must be set:

```
com.sun.am.policy.agents.fqdnMap
```

EXAMPLE 6–4 Property Settings: Load Balancer in Front of Web Agent

```
com.sun.am.policy.agents.fqdnMap = valid|LB-hostname
```

where *LB-hostname* represents the name of the machine on which the load balancer is located.

However, if SSL-termination or a proxy server is used in the deployment, all the following properties in the web agent `AMAgent.properties` configuration file should be set in addition to the preceding property:

```
com.sun.am.policy.agents.config.override_protocol
com.sun.am.policy.agents.config.override_host
com.sun.am.policy.agents.config.override_port
com.sun.am.policy.agents.config.agenturi.prefix
```

This example illustrates how properties can be set to enable load balancing when the protocol, hostname, and port number of the load balancer differ from that of the web agent. However, if the load balancer and the web agent share one of these characteristics, such as the protocol or hostname, then the respective property would be left blank instead of being assigned a value of *true*.

```
com.sun.am.policy.agents.config.override_protocol = true
com.sun.am.policy.agents.config.override_host = true
com.sun.am.policy.agents.config.override_port = true
com.sun.am.policy.agents.config.agenturi.prefix = LB-url/amagent
```

where *LB-url* represents the load balancer URL. The following example is a conceivable load balancer URL:

```
http://hostname.example.com:8080
```

Load Balancers in Front of Both the Web Agent and Access Manager

This scenario is simply a combination of the scenarios described in the preceding sections. See "Load Balancer in Front of Access Manager" on page 96 and "Load Balancer in Front of Web Agent" on page 97.

Key Features and Tasks Performed With Web Agent Scripts in Policy Agent 2.2

This section simply summarizes the types of scripts you can use with a web agent in Policy Agent 2.2. Refer to the relevant sections of this guide for specific information about the tasks performed with scripts. Scripts are used in performing the following tasks:

- Installing the initial web agent
  - Installation script for Solaris systems: `setup`
  - Installation script for AIX systems: `setup`
  - Installation script for Linux systems: `setup`
  - Installation script for Windows systems: `cscript.exe`

    which requires both of the following:
    - `ApacheCreateConfig.vbs`
    - `ApacheAdmin.vbs`
- Resetting the Shared Secret
  - Encryption script for Solaris systems: `crypt_util`
  - Encryption script for AIX systems: `crypt_util`
  - Encryption script for Linux systems: `crypt_util`
  - Encryption command for Windows systems: `cryptit`
- Uninstalling the initial web agent
  - Uninstallation script for Solaris systems: `uninstall_agent`
  - Uninstallation command for AIX systems:
    `java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent`
  - Uninstallation script for Linux systems: `uninstall_linux_agent_apache`
  - Uninstallation command for Windows systems: Requires two commands
    - `cscript.exe` using the `ApacheAdmin.vbs` script
    - `java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent`

After a web agent is installed, most interactions with the agent are performed by editing the web agent `AMAgent.properties` configuration file. However, a few tasks as mentioned in the preceding list are performed with scripts.

◆ ◆ ◆ CHAPTER 7

Uninstalling Policy Agent 2.2 for Apache HTTP Server

This chapter first presents you with a method for disabling a web agent. The task is the same for all web agents on all platforms. Then the chapter leads you through the uninstallation process on the available platforms as follows:

All Systems: Disabling a Web Agent in Policy Agent 2.2

In certain situations, you might want to disable a web agent temporarily. You can disable any web agent by resetting the property that controls the not-enforced URI list in the web agent `AMAgent.properties` configuration file.

▼ To Disable a Web Agent in Policy Agent 2.2

This task requires that you reset the following property:

```
com.sun.am.policy.agents.config.notenforced_list
```

1. **Reset the value of this property to the asterisk, "*," as follows:**

   ```
   com.sun.am.policy.agents.config.notenforced_list = *
   ```

2. **Restart Apache HTTP Server.**

Solaris Systems: Agent Uninstallation for Apache HTTP Server

You can uninstall a web agent on a Solaris system using a graphical user interface (GUI) or using the command line. However, if you are going to uninstall the deployment container, make sure that you uninstall the web agent before you uninstall the deployment container.

GUI Uninstallation of Agent for Apache HTTP Server on Solaris Systems

To uninstall a web agent, you must run the uninstallation program.

▼ To Uninstall Agent for Apache HTTP Server on Solaris Systems Using the GUI

1. **In the Policy Agent base directory (*PolicyAgent-base*), enter the following command:**

   ```
   # ./uninstall_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 Apache HTTP Server instance from which you just uninstalled the agent.**

Command-Line Uninstallation of Agent for Apache HTTP Server on Solaris Systems

To uninstall an agent, you must run the uninstallation program.

▼ To Uninstall Agent for Apache HTTP Server on Solaris Systems Using the Command Line

1. **In the Policy Agent base directory (*PolicyAgent-base*), enter the following command:**

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

   The uninstallation program detects the agent that was previously installed using the `setup` program and displays the following text:

   ```
   Ready to Uninstall

   1. Uninstall Now
   ```

```
2. Start Over
3. Exit Uninstallation
```

2 **Enter `1` to uninstall the agent.**

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

The uninstallation program displays the following text:

```
Product                                          Result     More Information
1.  Sun Java(tm) System Access Manager Policy Agent  Full       Available
2.  Done
```

4 **To see log information, enter `1`. To exit the uninstallation program, enter `2`.**

5 **When the uninstallation is complete, restart the Apache HTTP Server instance from which you just uninstalled the agent.**

AIX Systems: Agent Uninstallation for Apache HTTP Server

You can uninstall a web agent on a AIX system using the command line. Keep the following in mind:

- If you want to uninstall the deployment container, make sure that you uninstall the web agent before you uninstall the deployment container.
- Set `LIBPATH` to include the `libpasswd.so` file.

  For more information about setting `LIBPATH`, see "Installation of Agent for Apache HTTP Server on AIX Systems" on page 38.

Command-Line Uninstallation of Agent for Apache HTTP Server on AIX Systems

To uninstall this web agent using the command line, you must run the uninstallation program as explained in the following task description.

▼ To Uninstall Agent for Apache HTTP Server on AIX Systems Using the Command Line

1. **Using the command line, change directories to the home directory.**
2. **Issue the following command:**

```
java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent -nodisplay
```

Linux Systems: Agent Uninstallation for Apache HTTP Server

You can uninstall a web agent on a Linux system using a graphical user interface (GUI) or using the command line.

GUI Uninstallation of Agent for Apache HTTP Server on Linux Systems

To uninstall a web agent, you must run the uninstallation program.

▼ To Uninstall Agent for Apache HTTP Server on Linux Systems Using the GUI

1. **In the Policy Agent base directory (*PolicyAgent-base*), enter the following command:**

```
# ./uninstall_linux_agent_apache
```

2. **Click Next on Welcome panel.**
3. **Click Uninstall Now.**
4. **Click Close after uninstallation is complete.**
5. **Restart the Apache HTTP Server instance from which you just uninstalled the agent.**

Next Steps A post-uninstallation task is required when Agent for Apache HTTP Server is used on SUSE Linux. See "Agent for Apache HTTP Server on SUSE Linux: Removing the `common-2.2` Package" on page 104.

Command-Line Uninstallation of Agent for Apache HTTP Server on Linux Systems

To uninstall a web agent, you must run the uninstallation program.

▼ To Uninstall Agent for Apache HTTP Server on Linux Systems Using the Command Line

1 **In the Policy Agent base directory (*PolicyAgent-base*), enter the following command:**

```
# ./uninstall_linux_agent_apache -nodisplay
```

The uninstallation program displays the following text:

```
Ready to Uninstall

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

2 **Enter `1`, to remove the product.**

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

The uninstallation program displays the following text:

```
Product                                          Result     More Information
1.  Sun Java(tm) System Access Manager 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 Apache HTTP Server instance from which you just uninstalled the agent.**

Next Steps A post-uninstallation task is required when Agent for Apache HTTP Server is used on SUSE Linux. See the following task.

Agent for Apache HTTP Server on SUSE Linux: Removing the common-2.2 Package

The common-2.2 package was installed as a post-installation step as described in "Agent for Apache HTTP Server on SUSE Linux: Obtaining the Required Libraries" on page 73. If you uninstall the agent, remove the package as described in the following task.

▼ To Remove the common-2.2 Package

- **Issue the following command:**

```
rpm -e common-2.2
```

Windows Systems: Agent Uninstallation for Apache HTTP Server

This the section leads you through the uninstallation process, which first requires you to unconfigure the agent from each web site for which it is currently configured. This section is organized as follows:

Unconfiguring Agent for Apache HTTP Server on Windows Systems

If you no longer require Agent for Apache HTTP Server to protect a particular web site, you can unconfigure the agent from that web site. Furthermore, if you want to uninstall the agent, you must first unconfigure it from all the web sites for which it was configured.

Perform the following steps to unconfigure Agent for Apache HTTP Server 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.

▼ To Unconfigure Agent for Apache HTTP Server on Windows Systems

1. **Stop the web site for which you have configured the agent.**
2. **Change to the directory** *PolicyAgent-base*\apache\bin
3. **Run the following VB script to unconfigure the agent:**

```
cscript.exe ApacheAdmin.vbs -unconfig defaultConfig
```

`ApacheAdmin.vbs`	is a VB script that uses the output of the `ApacheCreateConfig.vbs` script. The output was saved to a configuration file, which for this example is represented by *defaultConfig*.
`-unconfig`	is the option that allows the output to be used to unconfigure the web site.
defaultConfig	represents the agent configuration file created previously as described in "Windows Systems: To Create Configuration Files, Agent for Apache HTTP Server" on page 52.

The script unconfigures the agent and displays the following message:

```
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 2.0.x. 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 HTTP Server.

Uninstallation of Agent for Apache HTTP Server on Windows Systems

Before running the uninstallation program, ensure that you have already unconfigured the agent from all the web sites for which it was configured as described in "Unconfiguring Agent for Apache HTTP Server on Windows Systems" on page 104. Perform the following steps to uninstall the agent.

▼ To Uninstall Agent for Apache HTTP Server

1. **Change to the following directory:**
   PolicyAgent-base\
2. **Run the following command to uninstall the agent:**
   ```
   java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent
   ```
3. **Click Next in the Welcome panel.**
4. **Click Uninstall Now.**
   The program uninstalls the agent.
5. **Restart the server.**

◆ ◆ ◆ APPENDIX A

Silent Installation of a Web Agent in Policy Agent 2.2

In addition to a standard installation of web agents, you can perform a silent installation as described in this appendix. The tasks involved in a silent installation of a web agent in Policy Agent 2.2 are similar for UNIX-based systems, which includes Linux systems, as described in the sections that follow. Silent installation is not available for Windows systems.

About Silent Installation of a Web Agent in Policy Agent 2.2

A silent installation refers to installing a program by implementing a script. The script is part of a state file. The script provides all the answers that you would normally supply to the installation program interactively. Running the script saves time and is useful when you want to install multiple instances of a web agent using the same parameters in each instance.

Silent installation is a simple two-step process of generating a state file and then using that state file. To generate a state file, you record the installation process, entering all the required information that you would enter during a standard installation. Then you run the installation program with the state file as the input source.

You can perform the tasks for a silent installation through the GUI (except on AIX systems) or through the command line as described in the respective sections that follow.

UNIX-based Systems: Silent Installation of a Web Agent in Policy Agent 2.2

The tasks that follow apply to Solaris systems and Linux systems.

Generating a State File for a Web Agent Installation on UNIX-based Systems

This section describes how to generate a state file for installing a web agent on UNIX-based systems, such as Solaris systems, AIX systems, and Linux systems. The description that follows provides an option of performing this task through the GUI, which does not apply to AIX systems, and an option of performing this task through the command line.

Regardless of which type of installation you choose, GUI or command line, you need to initially issue one command for recording the information you will enter as you follow the agent installation steps. Enter all the necessary installation information in order to create a complete state file.

▼ To Generate a State File for a Web Agent Installation on UNIX-based Systems

The following task describes how to generate a state file for a web agent installation on a Solaris system or a Linux system.

1 Change to the directory in which you unpacked the agent binaries.

This directory contains the `setup` program, which is used for installing a web agent and for performing other tasks.

2 Issue the command that applies as follows:

❒ To use the GUI installation (not applicable to AIX systems), issue the following command:

```
# ./setup -saveState filename
```

❒ To use the command-line installation, issue the following command:

```
# ./setup -nodisplay -saveState filename
```

`-saveState` An option that saves all of your responses to installation prompts in a state file.

filename Represents the name that you choose for the state file.

3 Enter the installation information as described in this guide.

See the appropriate section of this guide, according to your installation requirements:

- GUI installation on Solaris Systems
- Command-Line installation on Solaris Systems
- Command-Line installation on AIX Systems
- GUI installation on Linux Systems
- Command-Line installation on Linux Systems

Your answers to the prompts are recorded in the state file. When the installation is complete, the state file is created in the same directory where the installation program is located.

Note – When generated, a state file will have `read` permissions for all users. However, because the state file contains clear text passwords, change the file permissions to restrict `read` and `write` access to the user `root`.

Using a State File for a Web Agent Silent Installation on UNIX-based Systems

This section describes how to use a state file for installing a web agent on Solaris systems and Linux systems.

▼ To Install a Web Agent Using a State File on UNIX-based Systems

To perform a silent installation of a web agent using a state file, perform the following:

1 Change to the directory in which you unpacked the agent binaries.

At this point, this directory should contain, amongst other items, the `setup` program and the web agent installation state file.

2 Issue the following command:

```
# ./setup -nodisplay -noconsole -state filename
```

`-state` An option that directs the installer to run in non-interactive mode as it obtains all responses to prompts from the named state file.

filename Represents the name of the state file from which the installer obtains all responses.

The installation takes place hidden from view. After completion, the program exits automatically and displays the prompt.

Note – Even though the silent installation does not validate the keys in the state file, avoid editing the values of the keys in the state file because the `setupSDK` script might report a corrupt state file when used during subsequent silent installations.

◆ ◆ ◆ APPENDIX B

Troubleshooting a Web Agent Deployment

This appendix applies to Agent for Apache HTTP Server. If a problem is discussed in this appendix, it either applies only to this agent or it applies to two or more agents with one of them being this agent. This appendix explains how you can resolve problems that you might encounter while deploying or using this web agent. Be sure to also check the *Sun Java System Access Manager Policy Agent 2.2 Release Notes*, to see if the problem that you encounter is a known limitation of the web agent. If workarounds are available for such problems, they will be provided in the release notes.

In this chapter, refer to the troubleshooting section applicable to your platform as follows:

Solaris Systems: Troubleshooting Symptoms in Agent for Apache HTTP Server

This section includes various problems you might encounter on Solaris systems. The explanation of the problem is followed by possible solutions.

Solaris Systems: Troubleshooting Symptom 1

Symptom: Cannot install the web agent after a previous installation has been removed.

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

Sun Java(tm) System; Access Manager Policy Agent for Apache 1.3.29 or 2.0.48 or 2.0.50 or 2.0.52 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 web agent.
- You might have a previously-installed web agent and did not use the web agent's uninstallation program to uninstall the agent.
- The installation program's `productregistry` file might be corrupted.

Possible Solutions: Performing the following troubleshooting activities might resolve the issue:

- Check that you have uninstalled any existing installation of the web agent.
- The `productregistry` file may be corrupted if there is no existing installation of the web 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 `productregistry` file before you make changes.

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

```
<compid>SUNWamapc
        <compversion>2.2
                <uniquename>SUNWamapc</uniquename>
                <vendor></vendor>
                <compinstance>1
                        <parent>Agent for Apache
                                <instance>1
                                        <version>2.2</version>
                                </instance>
                        </parent>
                        <comptype>COMPONENT</comptype>
                        <location>/opt/apache_agent</location>
                        <dependent>
                                <compref>Agent for Apache
                                        <instance>1
                                                <version>2.2</version>
                                        </instance>
                                </compref>
                        </dependent>
                        <data>
                                <key>pkgs
                                        <value>SUNWamapc</value>
                                </key>
```

```
                        </data>
                    </compinstance>
            </compversion>
    </compid>
    <compid>Agent for Apache
            <compversion>2.2
                    <uniquename>Agent for Apache</uniquename>
                    <vendor></vendor>
                    <compinstance>1
                        <parent>Sun Java(tm) System Access Manager Policy Agent
                                    <instance>1
                                            <version>2.2</version>
                                    </instance>
                        </parent>
                            <children>
                                    <compref>SUNWamapc
                                            <instance>1
                                                    <version>2.2</version>
                                            </instance>
                                    </compref>
                            </children>
                            <comptype>FEATURE</comptype>
                            <location>/opt/apache_agent</location>
                            <dependent>
                             <compref>Sun Java(tm) System Access Manager Policy Agent
                                            <instance>1
                                                    <version>2.2</version>
                                            </instance>
                              </compref>
                            </dependent>
                            <required>
                                    <compref>SUNWamapc
                                            <instance>1
                                                    <version>2.2</version>
                                            </instance>
                                    </compref>
                            </required>
             </compinstance>
            </compversion>
    </compid>
    <compid>Sun Java(tm) System Access Manager Policy Agent
            <compversion>2.2
             <uniquename>Sun Java(tm) System Access Manager Policy Agent</uniquename>
             <compinstance>1
                            <children>
                                    <compref>Agent Utils
                                            <instance>2
                                                    <version>2.2</version>
```

```
                                                </instance>
                                        </compref>
                                        <compref>Agent for Apache
                                                <instance>1
                                                        <version>2.2</version>
                                                </instance>
                                        </compref>
                                </children>
                                <comptype>PRODUCT</comptype>
                                <location>/opt/apache_agent</location>
                                <uninstaller>/usr/java/bin/java -classpath
/opt/apache_agentuninstall_Sun_Java_tm__System_Access_Manager_Policy_Agent</uninstaller>
                                <required>
                                        <compref>Agent Utils
                                                <instance>2
                                                        <version>2.2</version>
                                                </instance>
                                        </compref>
                                        <compref>Agent for Apache
                                                <instance>1
                                                        <version>2.2</version>
                                                </instance>
                                        </compref>
                                </required>
                        </compinstance>
                </compversion>
        </compid>
```

Solaris Systems: Troubleshooting Symptom 2

Symptom: The uninstallation program does not remove entries from the agent's web container.

Possible Causes: Another instance of the web agent exists that was configured using the configuration script.

Possible Solution: Remove all the instances of the web agent using the `unconfig` script before running the uninstallation program.

Solaris Systems: Troubleshooting Symptom 3

Symptom: The browser goes into a loop for approximately a minute before displaying an access-denied page.

Possible Cause: The user tries to access a resource for which a policy with a time condition has been set and the time on the web agent host and the Access Manager host are not in sync.

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

Solaris Systems: Troubleshooting Symptom 4

Symptom: This problem is specific to Agent for Apache HTTP Server. The following error message is encountered:

```
The directory you provided does not contain a httpd
binary<p> file.  Please re-enter the full path to the<p>
directory where the Apache httpd binary file is located
```

This error message can occur during installation of Agent for Apache HTTP Server after you provide information about Apache Binary Directory. The following is an example of a directory name you might provide, which would normally not result in an error message:

```
/usr/apache/bin
```

Possible Cause

The version of Apache HTTP Server that you are using might be the version that comes bundled with Solaris™ 9 Operating System or with Solaris 10 Operating System. These Apache HTTP Server bundled packages are incomplete and should not be used. For example, such bundled packages do not come with the `httpd.conf` file to which the preceding error message refers.

Possible Solution: Download the desired version of Apache HTTP Server from the Apache web site at `http://www.apache.org/`. Compile and install the downloaded version of Apache HTTP Server before attempting to install the agent.

Solaris Systems: Troubleshooting Symptom 5

Symptom: When a user attempts to access a resource using Internet Explorer as the browser, access is denied.

Possible Cause: Internet Explorer overrides the port number of the web agent with the Access Manager port number. In such cases, the agent log file lists the URL that is being evaluated. The port number for that URL is incorrect.

Possible Solution: You can ensure this problem does not occur by setting the following property in the web agent `AMAgent.properties` configuration file to `true` as shown:

```
com.sun.am.policy.agents.config.override_port = true
```

AIX Systems: Troubleshooting Symptoms in Agent for Apache HTTP Server

This section includes various problems you might encounter with this agent on AIX systems. The symptom of the problem is followed by possible causes and solutions.

AIX Systems: Troubleshooting Symptom 1

Symptom: The browser goes into a loop for approximately a minute before displaying an access-denied page.

Possible Cause: The user tries to access a resource for which a policy with a time condition has been set and the time on the web agent host and the Access Manager host are not in sync.

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

AIX Systems: Troubleshooting Symptom 2

Symptom: The agent goes into an infinite loop.

Possible Cause: The value for the following property in the web agent `AMAgent.properites` configuration file is a resource to which users are assigned:

```
com.sun.am.policy.agents.config.accessdenied.url
```

The users assigned to this resource, do not have `allow` in the policy definition.

Possible Solution: For the `get` method, specify `allow` in the policy definition.

AIX Systems: Troubleshooting Symptom 3

Symptom: When a user attempts to access a resource using Internet Explorer as the browser, access is denied.

Possible Cause: Internet Explorer overrides the port number of the web agent with the Access Manager port number. In such cases, the agent log file lists the URL that is being evaluated. The port number for that URL is incorrect.

Possible Solution: You can ensure this problem does not occur by setting the following property in the web agent `AMAgent.properties` configuration file to `true` as shown:

```
com.sun.am.policy.agents.config.override_port = true
```

Linux Systems: Troubleshooting Symptoms in Agent for Apache HTTP Server

This section includes various problems you might encounter on Linux systems. The explanation of the problem is followed by possible solutions.

Linux Systems: Troubleshooting Symptom 1

Symptom: When a user attempts to access a resource using Internet Explorer as the browser, access is denied.

Possible Cause: Internet Explorer overrides the port number of the web agent with the Access Manager port number. In such cases, the agent log file lists the URL that is being evaluated. The port number for that URL is incorrect.

Possible Solution: You can ensure this problem does not occur by setting the following property in the web agent `AMAgent.properties` configuration file to `true` as shown:

```
com.sun.am.policy.agents.config.override_port = true
```

Windows Systems: Troubleshooting Symptoms in Agent for Apache HTTP Server

This section includes various problems you might encounter on Windows systems. The explanation of the problem is followed by possible solutions.

Windows Systems: Troubleshooting Symptom 1

Symptom: Cannot install the web agent after a previous installation has been removed.

Possible Causes:

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

Possible Solution: To resolve the issue, manually remove the web agent as explained in the following task description.

▼ To Manually Remove Agent for Apache HTTP Server

1. **Stop all of the web sites.**
2. **Stop the web server instance.**
3. **Remove Agent for Apache HTTP Server.**
   a. **In the Start menu, select Control Panel->Add/Remove programs**
   b. **Select Sun Java System Access Manager Policy Agent** *PolicyAgent-base***.**
   c. **Click Remove**
4. **Remove the** *PolicyAgent-base* **directory from the server.**
   where *PolicyAgent-base* represents the directory in which the web agent was originally installed.
5. **Remove the following entries from the** PATH **variable:**
   - *PolicyAgent-base*`\bin`
   - *PolicyAgent-base*`\es6\bin`
6. **Restart the server.**

Windows Systems: Troubleshooting Symptom 2

Symptom: Unable to uninstall the agent from a Windows system using the Add/Remove Program option in the Control Panel.

Possible Causes: Java's class path might not be set correctly on the machine.

Possible Solution: Perform the following task.

▼ To Uninstall a Web Agent on a Windows System When the GUI Uninstallation Fails

1. **Open Command Prompt Window.**
2. **Change directories to** *PolicyAgent-base*
3. **Execute the following command:**

```
java uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent
```

Windows Systems: Troubleshooting Symptom 3

Symptom: When a user attempts to access a resource using Internet Explorer as the browser, access is denied.

Possible Cause: Internet Explorer overrides the port number of the web agent with the Access Manager port number. In such cases, the agent log file lists the URL that is being evaluated. The port number for that URL is incorrect.

Possible Solution: You can ensure this problem does not occur by setting the following property in the web agent `AMAgent.properties` configuration file to `true` as shown:

```
com.sun.am.policy.agents.config.override_port = true
```

◆ ◆ ◆ APPENDIX C

Web Agent AMAgent.properties Configuration File

The web agent AMAgent.properties configuration file contains the necessary configuration properties needed for the web agent to function properly. It also contains the necessary information needed for the Sun Java System Access Manager SDK to function properly in a client installation mode as used by the web agent.

Properties in the Web Agent AMAgent.properties Configuration File

The web agent AMAgent.properties configuration file is located as described in Table 6–1. For a more detailed discussion of the key tasks you can perform using this configuration file, see "Key Features and Tasks Performed with the Web Agent AMAgent.properties Configuration File" on page 81.

For detailed information about every property, see the actual web agent AMAgent.properties configuration file in the product itself for a description of each property.

Most property names in the web agent AMAgent.properties configuration file have changed for Policy Agent 2.2. The following list highlights the change in property names by presenting the current property name in the release paired with the former property name from the 2.1 release. You can use this information to map the former property name to the current property name. Most properties apply to all web agents in the 2.2 release. A few properties are specific to one or a few web agents.

TABLE C–1 Changes in the Web Agent AMAgent.properties Configuration File for Policy Agent 2.2

2.2 Property Name	Former Property Name: 2.1 and Prior
com.sun.am.cookie.name	com.sun.am.cookieName

TABLE C–1 Changes in the Web Agent AMAgent.properties Configuration File for Policy Agent 2.2 *(Continued)*

2.2 Property Name	Former Property Name: 2.1 and Prior
com.sun.am.cookie.encode	com.sun.am.cookieEncoded
com.sun.am.log.level	com.sun.am.logLevels
com.sun.am.naming.url	com.sun.am.namingURL
com.sun.am.sslcert.dir	com.sun.am.sslCertDir
com.sun.am.certdb.prefix	com.sun.am.certDbPrefix
com.sun.am.trust_server_certs	com.sun.am.trustServerCerts
com.sun.am.notification.enable	com.sun.am.notificationEnabled
com.sun.am.notification.url	com.sun.am.notificationURL
com.sun.am.load_balancer.enable	com.sun.am.loadBalancer_enable
com.sun.am.policy.am.login.url	com.sun.am.policy.am.loginURL
com.sun.am.policy.am.username (unchanged)	com.sun.am.policy.am.username
com.sun.am.policy.am.password (unchanged)	com.sun.am.policy.am.password
com.sun.am.policy.am.url_comparison.case_ignore	com.sun.am.policy.am.urlComparison.caseIgnore
com.sun.am.policy.am.polling.interval	com.sun.am.policy.am.cacheEntryLifeTime
com.sun.am.policy.am.userid.param	com.sun.am.policy.am.userIdParam
com.sun.am.policy.am.lb.cookie.name	com.sun.am.policy.am.ias_SLB_cookie_name
com.sun.am.policy.am.fetch_from_root_resource	com.sun.am.policy.am.fetchFromRootResource
com.sun.am.policy.agents.config.local.log.file	com.sun.am.logFile
com.sun.am.policy.agents.config.local.log.rotate	NEW PROPERTY
com.sun.am.policy.agents.config.local.log.size	NEW PROPERTY
com.sun.am.policy.agents.config.remote.log	com.sun.am.serverLogFile
com.sun.am.policy.agents.config.profile.attribute.fetch.mode	com.sun.am.policy.am.ldapattribute.mode
com.sun.am.policy.agents.config.profile.attribute.map	com.sun.am.policy.am.headerAttributes

TABLE C–1 Changes in the Web Agent AMAgent.properties Configuration File for Policy Agent 2.2 *(Continued)*

2.2 Property Name	Former Property Name: 2.1 and Prior
com.sun.am.policy.agents.config. profile.attribute.cookie.prefix	com.sun.am.policy.am.ldapattribute. cookiePrefix
com.sun.am.policy.agents.config. profile.attribute.cookie.maxage	com.sun.am.policy.am.ldapattribute. cookieMaxAge
com.sun.am.policy.agents.config. session.attribute.fetch.mode	NEW PROPERTY
com.sun.am.policy.agents.config. session.attribute.map	NEW PROPERTY
com.sun.am.policy.agents.config. response.attribute.fetch.mode	NEW PROPERTY
com.sun.am.policy.agents.config. add_response_attrs	NEW PROPERTY
com.sun.am.policy.agents.config.version	com.sun.am.policy.agents.version
com.sun.am.policy.agents.config. audit.accesstype	com.sun.am.policy.agents.logAccessType
com.sun.am.policy.agents.config. agenturi.prefix	com.sun.am.policy.agents.agenturiprefix
com.sun.am.policy.agents.config.locale	com.sun.am.policy.agents.locale
com.sun.am.policy.agents.config. instance.name	com.sun.am.policy.agents.instanceName
com.sun.am.policy.agents.config. do_sso_only	com.sun.am.policy.agents.do_sso_only
com.sun.am.policy.agents.config. accessdenied.url	com.sun.am.policy.agents.accessDeniedURL
com.sun.am.policy.agents.config. url.redirect.param	com.sun.am.policy.agents.urlRedirectParam
com.sun.am.policy.agents.config. fqdn.default	com.sun.am.policy.agents.fqdnDefault
com.sun.am.policy.agents.config.fqdn.map	com.sun.am.policy.agents.fqdnMap
com.sun.am.policy.agents.config. cookie.reset.enable	com.sun.am.policy.agents. cookie_reset_enabled
com.sun.am.policy.agents.config. cookie.reset.list	com.sun.am.policy.agents.cookie_reset_list

TABLE C–1 Changes in the Web Agent AMAgent.properties Configuration File for Policy Agent 2.2 *(Continued)*

2.2 Property Name	Former Property Name: 2.1 and Prior
com.sun.am.policy.agents.config. cookie.domain.list	com.sun.am.policy.agents.cookieDomainList
com.sun.am.policy.agents.config. anonymous_user	com.sun.am.policy.agents. unauthenticatedUser
com.sun.am.policy.agents.config. anonymous_user.enable	com.sun.am.policy.agents. anonRemoteUserEnabled
com.sun.am.policy.agents.config. notenforced_list	com.sun.am.policy.agents. notenforcedList
com.sun.am.policy.agents.config. notenforced_list.invert	com.sun.am.policy.agents. reverse_the_meaning_of_notenforcedList
com.sun.am.policy.agents.config. notenforced_client_ip_list	com.sun.am.policy.agents. notenforced_client_IP_address_list
com.sun.am.policy.agents.config. postdata.preserve.enable	com.sun.am.policy.agents. is_postdatapreserve_enabled
com.sun.am.policy.agents.config. postcache.entry.lifetime	com.sun.am.policy.agents. postcacheentrylifetime
com.sun.am.policy.agents.config. cdsso.enable	com.sun.am.policy.agents.cdsso-enabled
com.sun.am.policy.agents.config. cdcservlet.url	com.sun.am.policy.agents.cdcservletURL
com.sun.am.policy.agents.config. client_ip_validation.enable	com.sun.am.policy.agents. client_ip_validation_enable
com.sun.am.policy.agents.config. logout.url	com.sun.am.policy.agents.logout.url
com.sun.am.policy.agents.config. logout.cookie.reset.list	com.sun.am.policy.agents.logout. cookie_reset_list
com.sun.am.policy.agents.config. get_client_host_name	com.sun.am.policy.agents.getClientHostname
com.sun.am.policy.agents.config. convert_mbyte.enable	com.sun.am.policy.agents. convertMbyteEnabled
com.sun.am.policy.agents.config. ignore_path_info	com.sun.am.ignore_path_info
com.sun.am.policy.agents.config. override_protocol	com.sun.am.policy.agents.overrideProtocol

TABLE C–1 Changes in the Web Agent AMAgent.properties Configuration File for Policy Agent 2.2 *(Continued)*

2.2 Property Name	Former Property Name: 2.1 and Prior
com.sun.am.policy.agents.config. override_host	com.sun.am.policy.agents.overrideHost
com.sun.am.policy.agents.config. override_port	com.sun.am.policy.agents.overridePort
com.sun.am.policy.agents.config. override_notification.url	com.sun.policy.agents. overrideNotificationUrl
com.sun.am.policy.agents.config. connection_timeout	NEW PROPERTY

◆ ◆ ◆ APPENDIX D

Error Codes

This appendix lists the error codes you might encounter while installing and configuring a web agent. It also provides explanations for the each code item.

Error Code List

This list of error codes includes locations that are reserved for error codes that do not currently exist.

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 Access Manager.
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.	Currently, no error code exists at this location.
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.	Currently, no error code exists at this location.
10. This is a reserved error code.	Currently, no error code exists at this location.
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.	Currently, no error code exists at this location.
14. AM_BUFFER_TOO_SMALL	The web agent does not have memory allocated to receive data from Access Manager.
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` error code. The error can occur in any of the API that take `am_policy_t` as a parameter.
16. AM_SERVICE_NOT_AVAILABLE	Currently, no error code exists at this location.
17. AM_ERROR_PARSING_XML	During communication with Access Manager, 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 Access Manager.
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 plug-in 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 Access Manager has failed. The details of this error can be found in the web agent's log file.

Index

T

U

V

W