Sun Java System Access Manager Policy Agent 2.2 Guide for IBM Lotus Domino 7.0

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

Part No: 820–1129–13
December 1, 2009

091201@23031

Contents

Preface

This Sun Java System Access Manager Policy Agent 2.2 Guide for IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0. For support and compatibility information about Agent for IBM Lotus Domino 7.0, see "Supported Platforms for the IBM Lotus Domino 7.0 Agent" on page 28.

Included in this guide is information about installing, configuring, uninstalling, and troubleshooting web agents, with the focus being on Policy Agent for IBM Lotus Domino 7.0.

Who Should Use This Book

This *Sun Java System Access Manager Policy Agent 2.2 Guide for IBM Lotus Domino 7.0* 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 IBM Lotus Domino 7.0," provides information specific to Policy Agent 2.2 for IBM Lotus Domino 7.0, focusing on aspects of the agent that make it unique compared to other web agents.

Chapter 3, "Installing the IBM Lotus Domino 7.0 Agent," provides instructions for installing Policy Agent 2.2 for IBM Lotus Domino 7.0.

Chapter 4, "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 IBM Lotus Domino 7.0," provides information about web agent configuration, some of which is required.

Chapter 6, "Managing Policy Agent 2.2 for IBM Lotus Domino 7.0," provides information about the methods available for managing Policy Agent 2.2 for IBM Lotus Domino 7.0, with most of the information being applicable to all web agents in the Policy Agent 2.2 software set.

Chapter 7, "Uninstalling the IBM Lotus Domino 7.0 Agent," provides instructions for uninstalling Policy Agent 2.2 for IBM Lotus Domino 7.0.

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 IBM Lotus Domino 7.0.

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 IBM Lotus Domino 7.0, 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

The following table lists documents in the documentation set and provides a description of each document.

Note – For instructions on installing Access Manager, see the *Sun Java Enterprise System 2005Q4 Installation Guide for UNIX*.

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 IBM Lotus Domino 7.0*, and the part number is 820-1129.

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 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 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

Starting with this release of web agents, when the current log file reaches a specific size, a new log file is created. Log information is then stored in the new log file until it reaches the size limit. This default behavior is configurable. Therefore, log rotation can be turned off and the size limit can be changed.

Note – The type of information stored in log files has not changed in Policy Agent 2.2. The following types of information are logged:

- Troubleshooting information
- Access denied information
- Access allowed information

The troubleshooting, or diagnostic, information is stored in log files, locally, with the web agent. The access denied and access allowed information, which is often referred to as audit-related information, can be stored both locally and with Access Manager.

Configuration that relates to the local log files is performed in the web agent `AMAgent.properties` configuration file. Configuration that relates to the audit related logs stored with Access Manager is performed in the Access Manager `AMConfig.properties` configuration file.

The log rotation described in this section refers to logs that store troubleshooting information locally.

Log rotation is controlled by the following configuration property in the web agent `AMAgent.properties` configuration file:

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

Log rotation occurs automatically since the default value of this property is `true`. When this property is set to `false`, no rotation takes place for the local log file.

The following example shows this configuration property set to `true`:

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

The following properties are also related to log rotation:

- The value for following configuration property indicates the location of the debug file:

  `com.sun.am.policy.agents.config.local.log.file`

- The value of following configuration property indicates the maximum number of bytes the debug file holds:

  `com.sun.am.policy.agents.config.local.log.size`

The following code example demonstrates how to set the property that controls log file size so that a new log file is created when the current log file reaches a specific size.

```
com.sun.am.policy.agents.config.local.log.size = n
```

Where *n* represents the size of a file in bytes. The file size should be a minimum of 3000 bytes. The default size is 10 megabytes.

Note – By default, the log file size property is not exposed in the web agent `AMAgent.properties` configuration file. If you want to change the default size, add a line to the file setting this property to the file size desired.

When a new log file is created an index appends to the name of the log file as such:

amAgent-1
amAgent-2

Where *amAgent* represents the fully qualified path name to the log files excluding the appended number. The numbers *1* and *2* represent the appended number. The appended number indicates the chronological order in which information of a given size was filed away into its respective log file. There is no limit to the number of log files that can be rotated.

Benefit - Log Rotation: Prior to this release of web agents, all logging messages were written to the same log file. However, saving all log information to a single log file has the potential of exhausting disk space. The log rotation feature solves this problem.

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 83

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 91.

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 Sun Java System Web Server 6.1 and an agent instance for Apache 2.0.52 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.

Using a Version 2.2 Policy Agent with OpenSSO Enterprise

Considerations for using a version 2.2 policy agent with Sun OpenSSO Enterprise (or Sun OpenSSO Express) include:

- OpenSSO Enterprise supports both version 3.0 and version 2.2 policy agents in the same deployment.
- A version 2.2 policy agent must continue to store its configuration data locally in its `AMAgent.properties` file. Therefore, because the version 2.2 policy agent configuration data is local to the agent, the OpenSSO centralized agent configuration option is not supported for version 2.2 agents. To configure a version 2.2 policy agent, you must continue to edit the agent's `AMAgent.properties` file.
- When you are configuring a version 2.2 policy agent with OpenSSO, the default Primary Server Deployment URI (and Failover Server Deployment URI, if required by the agent) is `/opensso` rather than `/amserver`.
- You can create a version 2.2 web agent or Java EE (formerly J2EE) agent profile in the OpenSSO Administration Console under Access Control, *realm-name*, Agents, and 2.2 Agents. However, you must do any additional configuration for the agent by editing its `AMAgent.properties` file.

◆ ◆ ◆ CHAPTER 2

About Policy Agent 2.2 for IBM Lotus Domino 7.0

This chapter provides information about Sun Java System Policy Agent 2.2 as it pertains specifically to IBM Lotus Domino 7.0, including:

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 System Access Manager Policy Agent 2.2 for IBM Lotus Domino 7.0, and that are unique to just the deployment container, IBM Lotus Domino 7.0. This chapter also summarizes specific tasks you might need to perform because of the unique characteristics of the deployment container.

Supported Platforms and Compatibility for the IBM Lotus Domino 7.0 Agent

Supported Platforms for the IBM Lotus Domino 7.0 Agent

TABLE 2–1 Supported Platforms for the IBM Lotus Domino 7.0 Agent

Agent For	Supported Platforms
IBM Lotus Domino 7.0 For the latest version of this agent, see the Sun Downloads page: `http://www.sun.com/download/index.jsp`	■ Solaris™ OS on SPARC® platforms, versions 8, 9, and 10 ■ Red Hat Enterprise Linux Advanced Server 3.0, 4.0, and 5.0, 32-bit only ■ IBM AIX 5L 5.1, 5.2, and 5.3 ■ Windows Server 2003, Enterprise Edition and Standard Edition

Minor versions of the IBM Lotus Domino 7.0 web container (such as 7.0.4) and minor versions of the supported platforms, including updates, service packs, and patches, are also supported.

Compatibility of the IBM Lotus Domino 7.0 Agent 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 26.

Information Specific to the IBM Lotus Domino 7.0 Agent

Note – The IBM Lotus Domino 7.0 agent and the IBM Lotus Domino 6.5.4 agent use the same agent binaries. However, in terms of deploying the agent, certain aspects differ between these two versions of IBM Lotus Domino. For example, the supported platforms and specific instructions can differ. Therefore, information about these two versions of IBM Lotus Domino are presented in two different guides. For information specific to Lotus Domino 6.5.4, see *Sun Java System Access Manager Policy Agent 2.2 Guide for IBM Lotus Domino 6.5.4*.

In this guide, you might see file or directory names that include the string `domino6`. Such references to `domino6` are correct, even though you are installing the agent on IBM Lotus Domino 7.0.

Support of Lotus Domino Database With the IBM Lotus Domino 7.0 Agent

You can configure the IBM Lotus Domino 7.0 agent to check if each user name that the agent authenticates exists in the Lotus Domino directory. A simple configuration step is required, which involves editing the web agent `AMAgent.properites` configuration file as described in "All Systems: Using the Lotus Domino Database for the IBM Lotus Domino 7.0 Agent" on page 50.

No Support of CDSSO With the IBM Lotus Domino 7.0 Agent

The version 2.2 for IBM Lotus Domino 7.0 agent does not support cross domain single sign-on (CDSSO). The IBM Lotus Domino 7.0 deployment container does not allow the agent to change the method type from `POST` to `GET`, which is necessary for cross domain single sign-on.

Support of Lightweight Third-Party Authentication (LTPA) With the IBM Lotus Domino 7.0 Agent

This technology for passing user authentication information between servers is supported by the IBM Lotus Domino 7.0 agent. For information on which properties in the web agent

`AMAgent.properties` configuration file affect the configuration of this technology, see "Configuring Agent for IBM Lotus Domino 7.0 with Lightweight Third-Party Authentication (LTPA)" on page 92.

◆ ◆ ◆ CHAPTER 3

Installing the IBM Lotus Domino 7.0 Agent

After you install the agent, continue with the post-installation tasks in Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for IBM Lotus Domino 7.0."

Installing the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

The following tasks apply to Solaris, Linux, and AIX systems:

Preparing to Install the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

▼ To Prepare to Install the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

1 **Ensure that the IBM Lotus Domino 7.0 agent is supported on the desired platform, as listed in "Supported Platforms and Compatibility for the IBM Lotus Domino 7.0 Agent" on page 27.**

2. **Install IBM Lotus Domino 7.0 server, if it is not already installed.**

   Refer to the IBM Lotus Domino 7.0 documentation for details about how to install and configure this server for your platform.

3. **Ensure that IBM Lotus Domino 7.0 has the latest patches available.**

4. **Set your `JAVA_HOME` environment variable to a JDK version 1.5 or later.**

   You must have JDK 1.5 or later to run the installation program graphical user interface (GUI).

   If you do not set the `JAVA_HOME` variable, the program will prompt you for the path to your JDK:

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

Installing the IBM Lotus Domino 7.0 Agent Using the GUI on UNIX and Linux Systems

▼ To Install the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

You must have superuser (`root`) permissions to run the web agent installation program.

1. **If necessary, download the IBM Lotus Domino 7.0 agent distribution file from the following site:**

   Sun Downloads: `http://www.sun.com/download/index.jsp`

2. **Unpack the distribution file in the directory of your choice. For example:**

   ```
   # gunzip -dc distribution-file.tar.gz| tar -xvof -
   ```

3. **In the same directory, issue the following command:**

   ```
   # ./setup
   ```

   The Welcome page appears.

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

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

6. **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*.

7 Click Next and provide the following information about the IBM Lotus Domino 7.0 instance the agent will protect:

Host Name: Enter the fully qualified domain name (FQDN) of the machine where the IBM Lotus Domino 7.0 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`.

Lotus Domino Data directory: Specify the IBM Lotus Domino 7.0 instance that this agent will protect. The following is the default Lotus Domino Data directory:

```
/local/notesdata
```

Web Server Port: Enter the port number for the IBM Lotus Domino 7.0 instance that will be protected by the web agent.

Web Server Protocol: If the IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0. 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 IBM Lotus Domino 7.0 instance where the agent is installed and *agent-deployment-uri* is the URI where the IBM Lotus Domino 7.0 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
```

---

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

9 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`.

If you are using the Lotus Domino 7.0 agent with Sun OpenSSO Enterprise or Sun OpenSSO Express, the URI should be `/opensso` rather than `/amserver`. For more information, see "Using a Version 2.2 Policy Agent with OpenSSO Enterprise" on page 26.

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.

If you are using the Lotus Domino 7.0 agent with Sun OpenSSO Enterprise or Sun OpenSSO Express, the URI should be `/opensso` rather than `/amserver`. For more information, see "Using a Version 2.2 Policy Agent with OpenSSO Enterprise" on page 26.

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, "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: Do not check this box. Cross domain single sign-on is not supported on Agent for IBM Lotus Domino 7.0. For more information see "Information Specific to the IBM Lotus Domino 7.0 Agent" on page 29.

10 After entering all the information, click Next.

11 **Review the installation summary to ensure that the information you have entered is correct.**

If you want to make changes, click Back. If all the information is correct, click Next.

12 **In the Ready to Install panel, click Install Now.**

13 **When the installation is complete, you can click Details to view details about the installation, or click Exit to end the installation program.**

14 **Restart the IBM Lotus Domino 7.0 instance on which you just installed the agent.**

Next Steps Continue with the post-configuration tasks, as described in Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for IBM Lotus Domino 7.0."

Installing the IBM Lotus Domino 7.0 Agent Using the Command-Line Interface (CLI) on UNIX and Linux Systems

▼ To Install the IBM Lotus Domino 7.0 Agent Using the Command-Line Interface (CLI) on UNIX and Linux Systems

1 **If necessary, download the IBM Lotus Domino 7.0 agent distribution file from the following site:**

Sun Downloads: `http://www.sun.com/download/index.jsp`

2 **Unpack the agent distribution file in the directory of your choice. For example:**

```
# gunzip -dc distribution-file.tar.gz| tar -xvof -
```

3 **In the directory in which you unpacked the agent distribution file, 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 IBM Lotus Domino 7.0 instance this agent will protect:**

- Host Name
- Lotus Domino Data directory
- Web Server Port
- Web Server Protocol
- Agent Deployment URI

For a description of the information to enter for these prompts, see "Installing the IBM Lotus Domino 7.0 Agent Using the GUI on UNIX and Linux Systems" on page 32.

6 **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 "Installing the IBM Lotus Domino 7.0 Agent Using the GUI on UNIX and Linux Systems" on page 32.

The following text is displayed:

```
Ready to Install

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

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

The following text is displayed:

```
Product                                          Result      More Information
1.  Sun 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 IBM Lotus Domino 7.0 instance on which you just installed the agent.**

Next Steps After you have installed the IBM Lotus Domino 7.0 agent, perform the applicable post-configuration tasks, as described in Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for IBM Lotus Domino 7.0."

Installing the IBM Lotus Domino 7.0 Agent on Windows Systems

Preparing to Install the IBM Lotus Domino 7.0 Agent 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 the IBM Lotus Domino 7.0 Agent on Windows Systems

1 **Ensure that Policy Agent 2.2 for IBM Lotus Domino 7.0 is supported on the desired platform as listed in "Supported Platforms and Compatibility for the IBM Lotus Domino 7.0 Agent" on page 27.**

2 **Install IBM Lotus Domino 7.0 server, if it is not already installed.**

Refer to the IBM Lotus Domino 7.0 documentation for details on how best to install and configure this server for your platform.

3 **Ensure that IBM Lotus Domino 7.0 has the latest patches available.**

4 **Set your `JAVA_HOME` environment variable to a JDK version 1.5 or later.**

You must have JDK 1.5 or later to run the graphical user interface (GUI) for the installation program.

If you not set the `JAVA_HOME` variable, the program will prompt you for the path to your JDK:

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

5 **Ensure that required libraries are available on the IBM Lotus Domino 7.0 instance.**

Depending on the Windows system you are using, the following libraries, `msvcp70.dll` and `msvcr70.dll`, might not be available. If these libraries are not available to the IBM Lotus Domino 7.0 instance , make them available as follows:

a. **Obtain these Windows libraries: `msvcp70.dll` and `msvcr70.dll`.**

These libraries come with certain Windows applications. You can also obtain them by contacting Sun technical support.

b. **Place the libraries in the `system32` subdirectory.**

The following path is an example of a conceivable path to this directory:

```
c:\WINDOWS\system32
```

Installing the IBM Lotus Domino 7.0 Agent on Windows Systems

On Windows systems, the web agent installation program has only the graphical user interface (GUI). You must have administrator privileges to run the installation program.

▼ To Install the IBM Lotus Domino 7.0 Agent on Windows Systems

1 **If necessary, download the IBM Lotus Domino 7.0 agent distribution file from the following site:**

Sun Downloads: `http://www.sun.com/download/index.jsp`

2 **If necessary, unzip the agent distribution file. For example:**

`unzip` *distribution-file*`.zip`

Note – On Microsoft Windows 2003, the zip file is not automatically unzipped. Therefore, after you download the 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.

---

3 **Run the installation program by double-clicking `setup.exe`.**

The Welcome page appears.

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

5 **Read the License Agreement. Click Yes to accept the license agreement.**

6 **Select the directory where 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*.

7 **Enter the applicable information about the IBM Lotus Domino 7.0 instance where this agent will be installed in the dialog box.**

The dialog box provides fields for entering the required information. You are prompted for information in the following order:

Host Name: Enter the fully qualified domain name (FQDN) of the system where the IBM Lotus Domino 7.0 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`.

Lotus Domino Data directory: Specify the IBM Lotus Domino 7.0 instance that this agent will protect. The following is the default Lotus Domino Data directory:

```
c:\Program Files\Lotus\Domino
```

The significance of this directory is that it contains the `notes.ini` file. The "Lotus Domino Data" wording is not a reference to any file or directory with "data" in the name.

Web Server Port: Enter the port number for the IBM Lotus Domino 7.0 instance that will be protected by the agent.

Web Server Protocol: If your IBM Lotus Domino 7.0 instance has been configured for SSL, then select HTTPS; otherwise select HTTP.

Agent Deployment URI: Enter a Universal Resource Identifier (URI) that will be used to access Agent for IBM Lotus Domino 7.0. 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 IBM Lotus Domino 7.0 instance where the agent is installed and *agent-deployment-uri* is the URI where the IBM Lotus Domino 7.0 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
```

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

9 Provide the following information about the Access Manager host:

The deployment container 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`.

If you are using the Lotus Domino 7.0 agent with Sun OpenSSO Enterprise or Sun OpenSSO Express, the URI should be `/opensso` rather than `/amserver`. For more information, see "Using a Version 2.2 Policy Agent with OpenSSO Enterprise" on page 26.

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.

If you are using the Lotus Domino 7.0 agent with Sun OpenSSO Enterprise or Sun OpenSSO Express, the URI should be `/opensso` rather than `/amserver`. For more information, see "Using a Version 2.2 Policy Agent with OpenSSO Enterprise" on page 26.

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, "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: Do not check this box. Cross domain single sign-on is not supported on Agent for IBM Lotus Domino 7.0. For more information see "Information Specific to the IBM Lotus Domino 7.0 Agent" on page 29.

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

11 **Review the installation summary to ensure that the information you have entered is correct.**

If you want to make changes, click Back. If all the information is correct, click Next.

12 **In the Ready to Install page, click Install Now.**

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

14 **Restart the IBM Lotus Domino 7.0 instance on which you just installed the agent.**

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

Next Steps After you have installed the IBM Lotus Domino 7.0 agent, perform the applicable post-configuration tasks, as described in Chapter 5, "Post-Installation Configuration: Policy Agent 2.2 for IBM Lotus Domino 7.0."

◆ ◆ ◆ CHAPTER 4

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

A web agent uses an agent profile to communicate with Access Manager server. The web agent uses the profile name and associated 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 the Access Manager Console and specify new 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, 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 the 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.

This section describes how to create or update an agent profile in the Access Manager Console and then how to make the corresponding changes in the web agent `AMAgent.properties` configuration file, including:

Note – If you want to change only the shared secret in the web agent and not the agent profile name, see "Resetting the Shared Secret Password" on page 88. A common reason to change only the shared secret is that it was entered incorrectly during the web agent installation.

Creating or Updating a Web Agent Profile in the Access Manager Console

▼ To Create or Update an Agent Profile in the Access Manager Console

The follow task describes how to create a new agent profile. If you are updating an existing agent profile, the steps are similar, except that you select an existing agent profile name in the Console.

1. **Log in to the Access Manager Admin Console.**
2. **Click Access Control and then the name of the realm for which you would like to create the agent profile.**
3. **Select Subjects and then Agent.**
4. **Click New and enter values for the following fields.**

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

   The agent uses this name to authenticate (with the following password) and communicate with Access Manager server. Multi-byte names are not accepted. Do not use the web agent default value of `UrlAccessAgent`.

   Password. Enter and confirm 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`.

   Device Status. The default status is Active, which allows the agent to authenticate and communicate with Access Manager server.

5. **Click Create.**

   The list of agents appears.

6 **(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 Web Agent Profile Name and Password

If you change the agent profile name and/or password in the Access Manager Console, you must assign the new value(s) to the corresponding properties in the web agent's `AMAgent.properties` configuration file.

Important. The values for the agent profile name and password must be the same for Access Manager server and in the web agent's `AMAgent.properties` configuration file

This task involves these basic steps:

1. If you changed the agent profile name in the Console, add the new name to the `com.sun.am.policy.am.username` property in the web agent's `AMAgent.properties` configuration file.
2. If you changed the agent profile password in the Console, encrypt the agent profile password using the encryption utility.
3. Add the new encrypted agent profile password from the previous step to the `com.sun.am.policy.am.password` property in the web agent's `AMAgent.properties` configuration file.

Follow these steps, depending on your platform:

- "To Update the Agent Profile Name and Agent Profile Password on UNIX and Linux Systems" on page 45
- "To Update the Agent Profile Name and Agent Profile Password on Windows Systems" on page 46

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

This task applies to Solaris, Linux, and AIX systems.

1. **If you changed the agent profile name in the Console, update the following property in the web agent's `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username=`*profile-name*

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

2. **If you changed the agent profile password in the Console, follow these steps:**

   a. **Change to the** *PolicyAgent-base*`/bin` **directory.**

   b. **Encrypt the agent profile password. For example:**

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

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

   c. **Copy the output from the `crypt_util` command and use it as the value for the following property in the web agent's `AMAgent.properties` configuration file:**

      `com.sun.am.policy.am.password=`*encrypted-password*

3. **Restart the IBM Lotus Domino 7.0 container.**

Next Steps To test the new password, try accessing a resource protected by the agent. If the agent is redirected to Access Manager, the password was changed properly.

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

1. **If you changed the agent profile name in the Console, update the following property in the web agent's `AMAgent.properties` configuration file:**

   `com.sun.am.policy.am.username=`*profile-name*

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

2. **If you changed the agent profile password in the Console, follow these steps:**

   a. **Change to the** *PolicyAgent-base*`/bin` **directory.**

   b. **Encrypt the agent profile password. For example:**

      `cryptit` *agent-profile-password*

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

c. Copy the output from the `cryptit` command and use it as the value for the following property in the web agent's `AMAgent.properties` configuration file:

`com.sun.am.policy.am.password=`*encrypted-password*

3 Restart the IBM Lotus Domino 7.0 container.

Next Steps To test the new password, try accessing a resource protected by the agent. If the agent is redirected to Access Manager, the password was changed properly.

◆ ◆ ◆ CHAPTER 5

Post-Installation Configuration: Policy Agent 2.2 for IBM Lotus Domino 7.0

This chapter describes configuration and other post-installation considerations and tasks for the IBM Lotus Domino 7.0 agent including:

For all supported platforms, configuring the DSAPI filter is a required task. After you configure the DSAPI filter, follow the guidelines for verifying that the installation was successful. This chapter is divided into the following sections:

As the preceding list indicates, many post-installation configuration tasks vary by platform. Refer to the platform-specific section that pertains to your site's deployment to perform the applicable tasks as described. The following list summarizes how various tasks apply to the supported platforms:

Using a Lotus Domino database with the IBM Lotus Domino 7.0 Agent
: This task applies to all supported platforms, but is only required when you want the agent to check the Lotus Domino database for a user name after the agent has authenticated that user name.

Setting file ownership and permissions
: This task is required on AIX systems for the agent to function. This task does not apply to Windows systems.

Setting `LIBPATH` to Required Agent Libraries
: This task is required on AIX systems for the agent to function.

Configuring the IBM Lotus Domino 7.0 DSAPI filter
: This task is required on all supported platforms for the agent to function.

Configuring the agent for multiple instances of IBM Lotus Domino 7.0
This task applies to AIX systems, but is only required when the agent is deployed on multiple instances of the IBM Lotus Domino 7.0 server.

Configuring SSL with the web agent
This task applies to all supported platforms, but is only required when you want to use SSL with the IBM Lotus Domino 7.0 agent.

After completing the applicable tasks described in this chapter and after verifying that the installation was successful, perform the applicable tasks to configure the web agent to your site's specific needs as explained in Chapter 6, "Managing Policy Agent 2.2 for IBM Lotus Domino 7.0."

All Systems: Using the Lotus Domino Database for the IBM Lotus Domino 7.0 Agent

You can configure the IBM Lotus Domino 7.0 agent to check the Lotus Domino database for each user name after the agent authenticates the user name. This configuration involves editing one property in the web agent `AMAgent.properties` configuration file as explained in the task description that follows.

The following property controls this feature:

```
com.sun.am.policy.agents.config.domino.check_name_database
```

The default setting for this property is `false`. When this property is set to `false`, after authenticating a user name, the agent does not check the Lotus Domino database for that user name. When this property is set to `true`, after authenticating a user name, the agent checks the Lotus Domino database for that user name. The following task description, shows this property set to `true`.

▼ To Configure the IBM Lotus Domino 7.0 Agent to Use the Lotus Domino Database

For this task you must edit the web agent `AMAgent.properties` configuration file.

- **Set the value of the following property to `true` as shown:**

```
com.sun.am.policy.agents.config.domino.check_name_database = true
```

Solaris Systems: Configuring the IBM Lotus Domino 7.0 Agent

After you check the file ownership and permissions (and reset if necessary), configure the DSAPI filter. Then, 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.

Solaris Systems: Setting File Ownership and Permissions for the IBM Lotus Domino 7.0 Agent

On Solaris systems, the IBM Lotus Domino 7.0 server must run as a non-root user. The default user created for this purpose during installation of the IBM Lotus Domino 7.0 server is `notes`. However, the actual user name will be different if this default was not accepted. For example purposes in this section, the default IBM Lotus Domino 7.0 user name of `notes` is used.

To enable the IBM Lotus Domino 7.0 agent to work properly, ensure that the `notes` user has read permissions to the following files:

- `/etc/opt/SUNWam/agents/domino/config/`*_PathInstanceName*`/AMAgent.properties`
- `/var/opt/SUNWam/agents/debug/`*_PathInstanceName*`/amAgent`
- *PolicyAgent-base*`/SUNWam/agents/domino/lib/libamdomino6.so`

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 Lotus Domino Data directory where slashes are converted to underscores. For this example, the path to the Lotus Domino Data directory is as follows:

```
/local/notesdata
```

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

```
_local_notesdata
```

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

```
chown notes:notes /etc/opt/SUNWam/agents/domino/config/_PathInstanceName
chown notes:notes /var/opt/SUNWam/agents/debug/_PathInstanceName/
chown notes:notes PolicyAgent-base/SUNWam/agents/domino/lib/libamdomino6.so
```

Additionally, if Access Manager is running in SSL mode, the files `cert7.db` and `key3.db` must also allow read access to the `notes` user. These files are available in the directory specified by the property `com.sun.am.sslcert.dir` in the web agent `AMAgent.properties` configuration file.

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

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

Solaris Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent

Configuring the DSAPI filter is a required task. When the DSAPI filter is not configured properly, users are unable to access resources. The DSAPI filter authenticates users and passes their information to the IBM Lotus Domino 7.0 server. The task description that follows explains how to configure the DSAPI filter.

▼ To Configure the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on Solaris Systems

If you want to configure the DSAPI filter for multiple server instances, repeat this task for each server instance that you want to support. Note that when multiple server instances exist, they all share a single web agent `AMAgent.properties` configuration file.

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**

   A window appears, presenting a list of servers.
3. **From the listed servers, select the IBM Lotus Domino 7.0 server instance that you want to configure.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**

6 **In the DSAPI Filter File Names field, enter the following file name:**

PolicyAgent-base/SUNWam/agents/domino/lib/libamdomino6.so

7 **Click the Save and Close button to save the changes.**

8 **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

Next Steps After you have configured the DSAPI filter, verify that the installation was successful. For information about the verification process, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 73.

Solaris Systems: Configuring the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances

To configure the IBM Lotus Domino 7.0 agent for multiple web server instances on a single computer, use the GUI or command-line version of the agent installation program to install the first agent. After the first agent is installed, you can then configure the agent for multiple instances of IBM Lotus Domino 7.0 using the `config` script that is copied into the system during the agent installation. This script must be run in the command line as described in the next section. The `config` script and the `unconfig` script are both located in the following directory:

PolicyAgent-base/SUNWam/agents/domino/bin

▼ To Configure the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances on Solaris Systems

1 **To configure the agent for additional IBM Lotus Domino 7.0 instances on a system, run the `config` script in the `bin` directory using the following command:**

```
# ./config
```

2 **Follow the prompts to install additional instances of the IBM Lotus Domino 7.0 Agent.**

For information on each of the prompts, see "Installing the IBM Lotus Domino 7.0 Agent Using the Command-Line Interface (CLI) on UNIX and Linux Systems" on page 35.

In general, information needs to be entered for both the protected IBM Lotus Domino 7.0 instance and the instances of Access Manager. The following text serves as example.

```
# ./config
Enter the Lotus Domino Data Directory:
Enter the Local Hostname:
Enter the Agent Web Server Port: [80]
Select Agent Web Server Protocol: [1] http [2] https --> [1]
Enter the Agent Deployment URI [/amagent]
Enter the Access Manager Hostname:
Enter the Access Manager Port: [58080]
Select Access Manager Protocol: [1] http [2] https --> [1]
Enter the Access Manager Deployment URI [/amserver]
Enter the Access Manager's Console Deployment URI [/amconsole]
Do You Want Failover Server Support: [1] yes [2] No --> [2]
Enter the User Name [UrlAccessAgent]
Enter Agent-Access Manager shared secret:
Re-enter Agent-Access Manager shared secret:
Is CDSSO Enabled: [1] yes [2] no --> [2]
Configuring webserver ...
Done.
```

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

Solaris Systems: Using SSL With the IBM Lotus Domino 7.0 Agent

During installation, if you choose the HTTPS protocol, the IBM Lotus Domino 7.0 agent is automatically configured and ready to communicate over Secure Sockets Layer (SSL). Before proceeding with tasks in this section, ensure that the IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0 server.

▼ To Configure Notifications for the IBM Lotus Domino 7.0 Agent for SSL on Solaris Systems

If IBM Lotus Domino 7.0 is running in SSL mode and is receiving notifications, first perform the following steps:

1. **Add the IBM Lotus Domino 7.0 certificate's root CA certificate to the Access Manager's certificate database.**

2. **Mark the CA root certificate as trusted to enable Access Manager to successfully send notifications to the IBM Lotus Domino 7.0 agent.**

Default Trust Behavior of the IBM Lotus Domino 7.0 Agent on Solaris Systems

This section only applies when Access Manager itself is running SSL. By default, the web agent installed on a remote IBM Lotus Domino 7.0 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 the IBM Lotus Domino 7.0 Agent on Solaris Systems" on page 55
- "Installing the Access Manager Root CA Certificate for a Remote IBM Lotus Domino 7.0 Instance on Solaris Systems" on page 56

Disabling the Default Trust Behavior of the IBM Lotus Domino 7.0 Agent 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 the IBM Lotus Domino 7.0 Agent 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 = /opt/domino/cert
   ```

3. **Set the Cert DB Prefix, if required.**

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

   ```
   com.sun.am.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 IBM Lotus Domino 7.0 Instance on Solaris Systems

The root CA certificate that you install on the remote instance of IBM Lotus Domino 7.0 must be the same certificate that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on IBM Lotus Domino 7.0 on Solaris Systems

The following steps outline a method for installing Access Manager Root CA Certificate on the IBM Lotus Domino 7.0 server. However, see the documentation for the IBM Lotus Domino 7.0 server for more information about installing certificates.

1. **(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/SUNWam/agents/domino/cert/certutil -N -d .
   ```

2. **Install the root CA certificate.**

   Remember that the root CA certificate that you install on the IBM Lotus Domino 7.0 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/SUNWam/agents/domino/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.`

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

PolicyAgent-base`/SUNWam/agents/domino/cert/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
```

4 Restart the IBM Lotus Domino 7.0 server.

AIX Systems: Configuring the IBM Lotus Domino 7.0 Agent

This section provides task descriptions for the following procedures:

- "AIX Systems: Setting File Ownership and Permissions for the IBM Lotus Domino 7.0 Agent" on page 58
- "AIX Systems: Setting `LIBPATH` to Include Libraries Specific to the IBM Lotus Domino 7.0 Agent" on page 59

- "AIX Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent" on page 60
- "AIX Systems: Configuring the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances" on page 61
- "AIX Systems: Using SSL With the IBM Lotus Domino 7.0 Agent" on page 62

After you check the file ownership and permissions (and reset if necessary), enable access to the proper libraries, configure the DSAPI filter, 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 for the IBM Lotus Domino 7.0 Agent

On AIX systems, the IBM Lotus Domino 7.0 server must run as a non-root user. The default user created for this purpose during installation of the IBM Lotus Domino 7.0 server is `notes`. However, the actual user name will be different if this default was not accepted. For example purposes in this section, the default IBM Lotus Domino 7.0 user name of `notes` is used.

To enable the IBM Lotus Domino 7.0 agent to work properly, ensure that the `notes` user has read permissions to the following files:

- `/etc/opt/agents/domino6/config/`*_PathInstanceName*`/AMAgent.properties`
- `/var/opt/agents/domino6/debug/`*_PathInstanceName*`/amAgent`
- *PolicyAgent-base*`/agents/domino6/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 Lotus Domino Data directory where slashes are converted to underscores. For this example, the path to the Lotus Domino Data directory is as follows:

```
/local/notesdata
```

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

```
_local_notesdata
```

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

```
chown notes:notes /etc/opt/agents/domino6/config/_PathInstanceName
chown notes:notes /var/opt/agents/domino6/debug/_PathInstanceName/
chown notes:notes PolicyAgent-base/agents/domino6/lib/libamdomino6.a
```

Additionally, if Access Manager is running in SSL mode, the files `cert7.db` and `key3.db` must also allow read access to the `notes` user. These files are available in the directory specified by the property `com.sun.am.sslcert.dir` in the web agent `AMAgent.properties` configuration file.

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

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

Where *cert-dir* represents the directory in which certificates and key stores related to SSL are located.

AIX Systems: Setting `LIBPATH` to Include Libraries Specific to the IBM Lotus Domino 7.0 Agent

On AIX systems, the libraries (version 3.9.5) and NSS (version 3.3.3) are not available by default, but are required.

The following task describes how to enable access to the required libraries.

▼ To set `LIBPATH` to Include Libraries Specific for the IBM Lotus Domino 7.0 Agent

1. **Download the following libraries:**

   NSPR (version 3.9.5) and NSS (version 3.3.3)

   These libraries can be downloaded from `http://www.mozilla.org`. You can then build the libraries on the AIX system.

2. **Before starting IBM Lotus Domino 7.0, set the LIBPATH environment variable.**

3. **Set the directory that contains the library `libamsdk.so` in the LIBPATH environment variable.**

   For example, if NSS and NSPR are located in the `/usr/mps` directory on the AIX system, you could issue the following command:

   ```
   setenv LIBPATH /PolicyAgent-base/agents/domino6/lib:/usr/mps:/lib:/usr/lib
   ```

AIX Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent

Configuring the DSAPI filter is a required task. When the DSAPI filter is not configured properly, users are unable to access resources. The DSAPI filter authenticates users and passes their information to the IBM Lotus Domino 7.0 server. The task description that follows explains how to configure the DSAPI filter.

▼ To Configure the DSAPI Filter With for the IBM Lotus Domino 7.0 Agent on AIX Systems

If you want to configure the DSAPI filter for multiple server instances, repeat this task for each server instance that you want to support. Note that when multiple server instances exist, they all share a single web agent `AMAgent.properties` configuration file.

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**

   A window appears, presenting a list of servers.
3. **From the listed servers, select the IBM Lotus Domino 7.0 server instance that you want to configure.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**
6. **In the DSAPI Filter File Names field, enter the following file name:**

   PolicyAgent-base`/agents/domino6/lib/libamdomino6.a`
7. **Click the Save and Close button to save the changes.**
8. **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

Next Steps After you have configured the DSAPI filter, verify that the installation was successful. For information about the verification process, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 73.

AIX Systems: Configuring the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances

To configure the IBM Lotus Domino 7.0 agent for multiple web server instances on a single computer, use the GUI or command-line version of the agent installation program to install the first agent. After the first agent is installed, you can then configure the agent for multiple instances of IBM Lotus Domino 7.0 using the `config` script that is copied into the system during the agent installation. This script must be run in the command line as described in the next section. The `config` script and the `unconfig` script are both located in the following directory:

PolicyAgent-base`/agents/domino6/bin`

▼ To Configure the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances on AIX Systems

1 **To configure the agent for additional IBM Lotus Domino 7.0 instances on a system, run the `config` script in the `bin` directory using the following command:**

```
# ./config
```

2 **Follow the prompts to install additional instances of the IBM Lotus Domino 7.0 agent.**

For information about each of the prompts, see "Installing the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems" on page 31.

In general, information needs to be entered for both the protected IBM Lotus Domino 7.0 instance and the instances of Access Manager. The following text serves as an example.

```
# ./config
Enter the Lotus Domino Data Directory:
Enter the Local Hostname:
Enter the Agent Web Server Port: [80]
Select Agent Web Server Protocol: [1] http [2] https --> [1]
Enter the Agent Deployment URI [/amagent]
Enter the Access Manager Hostname:
Enter the Access Manager Port: [58080]
Select Access Manager Protocol: [1] http [2] https --> [1]
Enter the Access Manager Deployment URI [/amserver]
Enter the Access Manager's Console Deployment URI [/amconsole]
Do You Want Failover Server Support: [1] yes [2] No --> [2]
Enter the User Name [UrlAccessAgent]
Enter Agent-Access Manager shared secret:
Re-enter Agent-Access Manager shared secret:
Is CDSSO Enabled: [1] yes [2] no --> [2]
```

```
Configuring webserver ...
Done.
```

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

AIX Systems: Using SSL With the IBM Lotus Domino 7.0 Agent

During installation, if you choose the HTTPS protocol, the IBM Lotus Domino 7.0 agent is automatically configured and ready to communicate over Secure Sockets Layer (SSL). Before proceeding with tasks in this section, ensure that the IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0 server.

▼ To Configure Notifications for the IBM Lotus Domino 7.0 Agent for SSL on AIX Systems

If IBM Lotus Domino 7.0 is running in SSL mode and is receiving notifications, first perform the following broadly defined steps:

1. **Add the IBM Lotus Domino 7.0 certificate's root CA certificate to the Access Manager's certificate database.**
2. **Mark the CA root certificate as trusted to enable Access Manager to successfully send notifications to the IBM Lotus Domino 7.0 agent.**

Default Trust Behavior of the IBM Lotus Domino 7.0 Agent on AIX Systems

This section only applies when Access Manager itself is running SSL. By default, the web agent installed on a remote IBM Lotus Domino 7.0 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 the IBM Lotus Domino 7.0 Agent 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 the IBM Lotus Domino 7.0 Agent 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 = /opt/domino/cert
   ```

3. **Set the Cert DB Prefix, if required.**

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

   ```
   com.sun.am.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 IBM Lotus Domino 7.0 Instance on AIX Systems

The root CA certificate that you install on the remote instance of IBM Lotus Domino 7.0 must be the same certificate that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on IBM Lotus Domino 7.0 on AIX Systems

The following steps outline a method for installing Access Manager Root CA Certificate on the IBM Lotus Domino 7.0 server. However, see the documentation for the IBM Lotus Domino 7.0 server for more information about installing certificates.

1 (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/agents/bin/certutil -N -d .
```

2 Install the root CA certificate.

Remember that the root CA certificate that you install on the IBM Lotus Domino 7.0 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/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 more information on the `cetrutil` utility enter `cetrutil -H for Help.`

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

```
PolicyAgent-base/agents/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
```

4 **Restart the IBM Lotus Domino 7.0 server.**

Windows Systems: Configuring the IBM Lotus Domino 7.0 Agent

This section provides task descriptions for the following procedures:

- "Windows Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent" on page 65
- "Windows Systems: Using SSL With the IBM Lotus Domino 7.0 Agent" on page 66

As described in the following subsections, configure the DSAPI filter. Then, perform the procedure for verifying a successful installation. Next, determine if the remaining procedure described in this section applies to your site's deployment scenario and perform the procedure if necessary.

Windows Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent

Configuring the DSAPI filter is a required task. When the DSAPI filter is not configured properly, users are unable to access resources. The DSAPI filter authenticates users and passes their information to the IBM Lotus Domino 7.0 server. The task description that follows explains how to configure the DSAPI filter.

▼ To Configure the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on Windows Systems

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**
   A window appears, presenting a list of servers.
3. **From the listed servers, select the IBM Lotus Domino 7.0 server instance that you want to configure.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**
6. **Click Edit Server.**
7. **In the DSAPI Filter File Names field, enter the following file name:**
   *PolicyAgent-base**domino**bin**amdomino6.dll*
8. **Click the Save and Close button to save the changes.**
9. **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

Next Steps After you have configured the DSAPI filter, verify that the installation was successful. For information about the verification process, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 73.

Windows Systems: Using SSL With the IBM Lotus Domino 7.0 Agent

During installation, if you choose the HTTPS protocol, the IBM Lotus Domino 7.0 agent is automatically configured and ready to communicate over SSL. Before proceeding with the tasks in this section, ensure that the IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0 server.

▼ To Configure Notifications for the IBM Lotus Domino 7.0 Agent for SSL on Windows Systems

If IBM Lotus Domino 7.0 is running in SSL mode and is receiving notifications, first perform the following broadly defined steps:

1. **Add the IBM Lotus Domino 7.0 certificate's root CA certificate to the Access Manager's certificate database.**
2. **Mark the CA root certificate as trusted to enable Access Manager to successfully send notifications to the IBM Lotus Domino 7.0 Agent.**

Default Trust Behavior of the IBM Lotus Domino 7.0 Agent on Windows Systems

This section only applies when Access Manager itself is running SSL. By default, theIBM Lotus Domino 7.0 agent 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 the IBM Lotus Domino 7.0 Agent on Windows Systems" on page 67
- "Installing the Access Manager Root CA Certificate on IBM Lotus Domino 7.0 on Windows Systems" on page 68

Disabling the Default Trust Behavior of the IBM Lotus Domino 7.0 Agent 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 the IBM Lotus Domino 7.0 Agent 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 IBM Lotus Domino 7.0 on Windows Systems

The root CA certificate that you install on the IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0 on Windows Systems

The following steps outline a method for installing Access Manager Root CA Certificate on the IBM Lotus Domino 7.0 server. However, see the documentation for the IBM Lotus Domino 7.0 server for more information about installing certificates.

1. **(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 .
   ```

2. **Install the root CA certificate.**

   Remember that the root CA certificate that you install on the IBM Lotus Domino 7.0 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.`

3. **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
```

4 **Restart the IBM Lotus Domino 7.0 server.**

Linux Systems: Configuring the IBM Lotus Domino 7.0 Agent

This section provides task descriptions for the following:

- "Linux Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent" on page 69
- "Linux Systems: Configuring the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances" on page 70
- "Linux Systems: Using SSL With the IBM Lotus Domino 7.0 Agent" on page 72

After you check the file ownership and permissions (and reset if necessary), configure the DSAPI filter. Then, 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.

Linux Systems: Configuring the DSAPI Filter for the IBM Lotus Domino 7.0 Agent

Configuring the DSAPI filter is a required task. When the DSAPI filter is not configured properly, users are unable to access resources. The DSAPI filter authenticates users and passes their information to the IBM Lotus Domino 7.0 server. The task description that follows explains how to configure the DSAPI filter.

▼ To Configure the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on Linux Systems

If you want to configure the DSAPI filter for multiple server instances, repeat this task for each server instance that you want to support. Note that when multiple server instances exist, they all share a single web agent `AMAgent.properties` configuration file.

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**

   A window appears, presenting a list of servers.
3. **From the listed servers, select the IBM Lotus Domino 7.0 server instance that you want to configure.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**
6. **In the DSAPI Filter File Names field, enter the following file name:**

   PolicyAgent-base`/agents/domino6/lib/libamdomino6.so`
7. **Click the Save and Close button to save the changes.**
8. **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

Next Steps After you have configured the DSAPI filter, verify that the installation was successful. For information about the verification process, see "All Systems: Verifying a Successful Installation on Policy Agent 2.2" on page 73.

Linux Systems: Configuring the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances

To configure the IBM Lotus Domino 7.0 agent for multiple web server instances on a single Linux computer, use the GUI or the command-line version of the agent installation program to install the first agent. After the first agent is installed, you can then install successive agents using the `config_linux` script. This script must be run in the command line as described in the next section. The `config_linux` script and the `unconfig_linux` script are both located in the following directory:

PolicyAgent-base/agents/domino6/bin

To Configure the IBM Lotus Domino 7.0 Agent on Multiple Web Server Instances on Linux Systems

Perform the following steps if you want to configure additional agents on a system after the original IBM Lotus Domino 7.0 agent has been installed.

1 **Run the `config_linux` script in the `bin` directory using the following command:**

```
# ./config_linux
```

2 **Follow the prompts to install additional instances of the IBM Lotus Domino 7.0 agent.**

For information about each of the prompts, see "Installing the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems" on page 31

In general, information needs to be entered for both the protected IBM Lotus Domino 7.0 instance and the instances of Access Manager. The following text serves as an example run:

```
# ./config_linux
Enter the Lotus Domino Data Directory:
Enter the Local Hostname:
Enter the Agent Web Server Port: [80]
Select Agent Web Server Protocol: [1] http [2] https --> [1]
Enter the Agent Deployment URI [/amagent]
Enter the Access Manager Hostname:
Enter the Access Manager Port: [58080]
Select Access Manager Protocol: [1] http [2] https --> [1]
Enter the Access Manager Deployment URI [/amserver]
Enter the Access Manager's Console Deployment URI [/amconsole]
Do You Want Failover Server Support: [1] yes [2] No --> [2]
Enter the User Name [UrlAccessAgent]
Enter Agent-Access Manager shared secret:
Re-enter Agent-Access Manager shared secret:
Is CDSSO Enabled: [1] yes [2] no --> [2]
Configuring webserver ...
Done.
```

Note – Be sure to use the `unconfig_linux` script to uninstall any agent that was installed using the `config_linux` script. You cannot use the GUI installation program to uninstall agents that were installed using the command line. The GUI uninstallation program must be executed only after unconfiguring all the existing agents installed using command-line `unconfig_linux` script.

Linux Systems: Using SSL With the IBM Lotus Domino 7.0 Agent

During installation, if you chose the HTTPS protocol, the IBM Lotus Domino 7.0 agent is automatically configured and ready to communicate over SSL. Before proceeding with the following tasks in this section, ensure that IBM Lotus Domino 7.0 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 IBM Lotus Domino 7.0 server.

▼ To Configure Notifications for the IBM Lotus Domino 7.0 Agent for SSL on Linux Systems

If IBM Lotus Domino 7.0 is running in SSL mode and is receiving notifications, first perform the following broadly defined steps:

1. **Add the IBM Lotus Domino 7.0 certificate's root CA certificate to the Access Manager's certificate database.**
2. **Mark the CA root certificate as trusted to enable Access Manager to successfully send notifications to the IBM Lotus Domino 7.0 agent.**

Default Trust Behavior of the IBM Lotus Domino 7.0 Agent on Linux Systems

This section only applies when Access Manager itself is running SSL. By default, the agent installed on a remote IBM Lotus Domino 7.0 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 the IBM Lotus Domino 7.0 Agent on Linux Systems" on page 72
- "Installing the Access Manager Root CA Certificate for a Remote IBM Lotus Domino 7.0 Instance on Linux Systems" on page 73

Disabling the Default Trust Behavior of the IBM Lotus Domino 7.0 Agent on Linux Systems

The following property in the web agent `AMAgent.properties` configuration file controls the agent's trust behavior, which 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 the IBM Lotus Domino 7.0 Agent 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 IBM Lotus Domino 7.0 Instance on Linux Systems

The root CA certificate that you install on the remote instance of IBM Lotus Domino 7.0 must be the same certificate that is installed on the Access Manager host.

▼ To Install the Access Manager Root CA Certificate on IBM Lotus Domino 7.0 on Linux Systems

- **For instructions on installing a root CA certificate on Linux systems, see the documentation for the IBM Lotus Domino 7.0 server.**

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 6

Managing Policy Agent 2.2 for IBM Lotus Domino 7.0

Interaction with Policy Agent 2.2 for IBM Lotus Domino 7.0 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:

- "Key Features and Tasks Performed with the Web Agent `AMAgent.properties` Configuration File" on page 75
- "Key Features and Tasks Performed With Web Agent Scripts or Commands in Policy Agent 2.2" on page 94

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 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 lists the default locations for the web agent AMAgent.properties configuration file.

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

Server	Platform	Location
IBM Lotus Domino 7.0	Solaris SPARC platform	/etc/opt/SUNWam/agents/domino/config/*_PathInstanceName*/
	Linux systems	/etc/opt/agents/domino6/config/*_PathInstanceName*/
	AIX Systems	/etc/opt/agents/domino6/config/*_PathInstanceName*/
	Windows systems	*PolicyAgent-base*\\domino\\config*_PathInstanceName*\\

where *_PathInstanceName* is derived from the full path to the Lotus Domino Data directory, which is the directory where data is stored for the IBM Lotus Domino 7.0 server. The *_PathInstanceName* directory is automatically created and named during the agent installation process.

The following are the default locations for the Lotus Domino Data directory depending on the platform:

Solaris Systems and Linux Systems:	`/local/notesdata`
AIX Systems:	`/local/notesdata`
Windows Systems:	`c:Program Files\Lotus\Domino`

The process of creating a name for the *_PathInstanceName* directory involves the conversion of slash symbols into underscore symbols.

For example, the preceding examples of Lotus Domino Data directory paths would be converted to the following *_PathInstanceName* directory names:

Solaris Systems and Linux Systems:	`_local_notesdata`
AIX Systems	`_local_notesdata`
Windows Systems:	`C_Program Files_Lotus_Domino`

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 a forward slash (`/`) to separate directories, not a backslash (`\`) or double backslashes (`\\`). This holds true even on Windows systems.
- Spaces in the Windows file names are allowed.

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 server 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 failoverAccess Manager host.
host3	Name of the second failoverAccess 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 and a cache notification 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.

Hybrid Cache Updates

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

Web agents have the hybrid cache update mode switched on by default. This is triggered by the property `com.sun.am.notification.enable` in the web agent `AMAgent.properties` configuration file, which is set to `true`. When the property is set to `false`, the web agent updates its cache through the cache polling mechanism only.

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

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

In a normal deployment situation, policy changes on the server are frequent, which requires sites to accept a certain amount of latency for 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

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

```
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.

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

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

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 in 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 appropriate XML file on the machine where Access Manager is installed. The following example demonstrates the appropriate XML file on Solaris systems.

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 is required 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

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, "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 platform-specific tasks (follow the steps according to the platform on which the agent is installed).

▼ To Reset the Shared Secret on Solaris Systems

1 Go to the following directory:

PolicyAgent-base`/bin`

2 Execute the following script in the command line:

`# ./crypt_util` *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 `crypt_util` 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.

To Reset the Shared Secret on Windows Systems

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

   PolicyAgent-base\\bin

2. **Execute the following script in the command line**

   `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 `cryptit` *shared-secret* 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.

To Reset the Shared Secret on Linux Systems

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

   PolicyAgent-base/bin

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

   `crypt_util` *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 `crypt_util` *shared-secret* 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 91
- "Load Balancer in Front of Web Agent" on page 91
- "Load Balancers in Front of Both the Web Agent and Access Manager" on page 92

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:

```
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 91 and "Load Balancer in Front of Web Agent" on page 91.

Configuring Agent for IBM Lotus Domino 7.0 with Lightweight Third-Party Authentication (LTPA)

This section applies to agents for IBM Lotus Domino.

LTPA is an authentication mechanism used by IBM Lotus Domino that provides users with single sign-on (SSO) capabilities between LTPA-technology-supported servers. If a deployment consists of servers that support LTPA technology, the LTPA token that is set as a cookie in the browser by the IBM Lotus Domino instance can be shared among servers. Thus, users are not prompted to enter their credentials (user name and password) every time they access a server.

You can edit various properties in the web agent `AMAgent.properties` configuration file to configure the LTPA mechanism to work with Agent for IBM Lotus Domino 7.0 as illustrated in the following list of properties:

- `com.sun.am.policy.agents.config.domino.ltpa.enable`

  Explanation: This property controls whether Agent for IBM Lotus Domino 7.0 uses an LTPA token or not.

  Possible Values: `true` and `false`.

  Default Value: `false`.

- `com.sun.am.policy.agents.config.domino.ltpa.cookie_name`

  Explanation: This property provides the name of the cookie that contains the LTPA token.

  Possible Values: *ltpa-cookie-name*, which represents the name of the cookie that contains the LTPA token.

  Default Value: `LtpaToken`.

- `com.sun.am.policy.agents.config.domino.ltpa.config_name`

  Explanation: This property provides the configuration name that Agent for IBM Lotus Domino 7.0 uses in order to employ the LTPA token mechanism. This property is similar to the preceding property concerning the cookie name in that it uses the same default value. The value set for this property is passed as a parameter during SSO token validation.

  Possible Values: *ltpa-configuration-name*, which represents the name of the configuration to which the LTPA token belongs.

  Default Value: `LtpaToken`.

- `com.sun.am.policy.agents.config.domino.ltpa.org_name`

  Explanation: This property provides the organization name to which the LTPA token belongs.

  Possible Values: *ltpa-cookie-organization-name*, which represents the organization to which the LTPA token belongs.

  Default Value: Null.

- `com.sun.am.policy.agents.config.domino.checkNameDatabase`

Explanation: This property is a flag that can be set to check whether or not the user exists in the IBM Lotus Domino 7.0 database. If the user exists, `REMOTE_USER` can then be set to the value specified in the database.

Possible Values: `true` and `false`.

Default Value: `false`.

Key Features and Tasks Performed With Web Agent Scripts or Commands in Policy Agent 2.2

This section simply summarizes the types of scripts or commands 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 commands or scripts. Commands or 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 command for Windows systems: `setup.exe`
  - Installation script for Linux systems: `setup`
- Configuring the web agent for multiple web server instances
  - Configuration script for Solaris systems: `config`
  - Configuration script for AIX systems: `config`
  - Configuration command for Windows systems: Not Applicable
  - Configuration script for Linux systems: `config_linux`
- Resetting the Shared Secret
  - Encryption script for Solaris systems: `crypt_util`
  - Encryption script for AIX systems: `crypt_util`
  - Encryption command for Windows systems: `cryptit`
  - Encryption script for Linux systems: `crypt_util`
- Unconfiguring the web agent for multiple web server instances
  - Unconfiguration script for Solaris systems: `unconfig`
  - Unconfiguration script for AIX systems: `unconfig`
  - Unconfiguration command for Windows systems: Not Applicable
  - Unconfiguration script for Linux systems: `unconfig_linux`
- Uninstalling the initial web agent
  - Uninstallation script for Solaris systems: `uninstall_agent`
  - Uninstallation script for AIX systems:
    `java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent`

- Uninstallation command for Windows systems:
  `java -cp . uninstall_Sun_Java_tm_System_Access_Manager_Policy_Agent`
- Uninstallation script for Linux systems: `uninstall_linux_agent_domino6`

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 commands or scripts.

◆ ◆ ◆ CHAPTER 7

Uninstalling the IBM Lotus Domino 7.0 Agent

Disabling a Version 2.2 Web Agent

This section applies to Solaris, Linux, and AIX systems.

In certain situations, you might want to temporarily disable a web agent. To disable a web agent, reset the `com.sun.am.policy.agents.config.notenforced_list` property to an asterisk (`*`) in the agent's `AMAgent.properties` configuration file. This property controls the not-enforced URI list for the agent.

▼ To Disable a Version 2.2 Web Agent

1 **In the web agent's `AMAgent.properties` configuration file, reset the `com.sun.am.policy.agents.config.notenforced_list` property to an asterisk (`*`). For example:**

```
com.sun.am.policy.agents.config.notenforced_list = *
```

2 **Restart the IBM Lotus Domino 7.0 instance.**

Uninstalling the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

This section applies to Solaris, Linux, and AIX systems.

You can uninstall a web agent on these systems using a graphical user interface (GUI) or command-line interface (CLI). However, first consider these items:

- If the agent was installed using the `config` script, run the `unconfig` script before you run the uninstallation program, as described in "Unconfiguring the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems" on page 98.
- Before you uninstall the IBM Lotus Domino 7.0 agent, remove the DSAPI filter, as described in "Removing the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems" on page 98.
- If you want to uninstall the IBM Lotus Domino 7.0 container, uninstall the agent before you uninstall the container.

Unconfiguring the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

To remove an instance of the IBM Lotus Domino 7.0 agent that was configured using the `config` script, you must run the `unconfig` script.

▼ To Unconfigure the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

- **Run the `unconfig` script. For example, on Solaris systems:**

```
# cd PolicyAgent-base/SUNWam/agents/domino/bin
# ./unconfig LotusDominoData-directory
Unconfiguring webserver ...
done.
```

where *LotusDominoData-directory* represents the path to the Lotus Domino Data directory. The default path is `/local/notesdata`.

Removing the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems

Before you uninstall the IBM Lotus Domino 7.0 agent, regardless of which uninstallation method you use (GUI or command line), first remove the DSAPI filter.

▼ To Remove the DSAPI Filter for the IBM Lotus Domino 7.0 on UNIX and Linux Systems

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**
   A window appears, presenting a list of servers.
3. **From the listed servers, select the server you want to uninstall.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**
6. **Remove the DSAPI filter file name specified for the agent, leaving the DSAPI Filter File Name field blank.**
7. **Click the Save and Close button to save the changes.**
8. **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

Uninstalling the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems Using the GUI

After you have removed the DSAPI filter, you can uninstall the agent.

▼ To Uninstall the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems Using the GUI

1. **In the *PolicyAgent-base* directory, enter the following command:**

```
# ./uninstall_agent
```

2. **Click Next on Welcome panel.**
3. **Click Uninstall Now on Ready to Uninstall panel.**
4. **After the uninstallation process is complete, click Close.**

5 **Restart the IBM Lotus Domino 7.0 instance.**

Uninstalling the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems Using the Command Line

After you have removed DSAPI Filter, you can uninstall the agent.

▼ To Uninstall the IBM Lotus Domino 7.0 Agent on UNIX and Linux Systems Using the Command Line

1 **In the *PolicyAgent-base* directory, 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 IBM Lotus Domino 7.0 instance.**

Uninstalling the IBM Lotus Domino 7.0 Agent on Windows Systems

On Windows systems, the uninstallation program has only the graphical user interface (GUI). You must have administrator privileges to run the program. First, remove the DSAPI filter and then uninstall the agent.

- "To Remove the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on Windows Systems" on page 101

■ "To Uninstall the IBM Lotus Domino 7.0 Agent on Windows Systems" on page 101

▼ To Remove the DSAPI Filter for the IBM Lotus Domino 7.0 Agent on Windows Systems

1. **In the Lotus Domino Administrator web console, select the Configuration tab.**
2. **In the left pane, under Server, click All Server Documents**
   A window appears, presenting a list of servers.
3. **From the listed servers, select the server you want to uninstall.**
4. **Click Internet Protocols.**
5. **Select the HTTP tab.**
6. **Remove the DSAPI filter file name specified for the agent, leaving the DSAPI Filter File Name field blank.**
7. **Click the Save and Close button to save the changes.**
8. **Open the IBM Lotus Domino 7.0 Quick Console and restart the server by entering the following commands:**

```
tell http quit
load http
```

▼ To Uninstall the IBM Lotus Domino 7.0 Agent on Windows Systems

1. **In the Start menu, choose Settings>Control Panel.**
2. **In the Control Panel, double click Add/Remove Programs**
3. **In the Add/Remove Programs window, choose Sun Java System Access Manager Policy Agent and click Change/Remove.**
4. **In the Welcome Panel, click Next.**
5. **In the Ready to Uninstall Panel, click Uninstall Now.**

6 **After the uninstallation process is complete, click Exit.**

7 **Restart the IBM Lotus Domino 7.0 instance.**

◆ ◆ ◆ 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 the same for Solaris systems and Linux systems, but different for Windows systems. This appendix provides the information for performing the respective tasks as follows:

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 or through the command line as described in the respective sections that follow.

Solaris and Linux 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 Solaris and Linux Systems

This section describes how to generate a state file for installing a web agent on Solaris systems and Linux systems. The description that follows provides an option of performing this task through the GUI 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 Solaris and Linux 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 following directory:

PolicyAgent-base/

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, 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 needs:

- GUI installation on Solaris Systems

- Command-Line installation on Solaris 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 Solaris and Linux 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 Solaris and Linux Systems

To perform a silent installation of a web agent using a state file, perform the following:

1 Change to the following directory:

PolicyAgent-base

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.

AIX Systems: Silent Installation of a Web Agent in Policy Agent 2.2

The tasks that follow apply to AIX systems.

Generating a State File for a Web Agent Installation on AIX Systems

This section describes how to generate a state file for installing a web agent on AIX systems. The description that follows provides an option of performing this task through the GUI 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 AIX Systems

The following task describes how to generate a state file for a web agent installation on AIX systems.

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, 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 needs:

- GUI installation on AIX Systems
- Command-Line installation on AIX 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 AIX Systems

This section describes how to use a state file for installing a web agent on AIX systems.

▼ To Install a Web Agent Using a State File on AIX 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.

Windows Systems: Silent Installation of a Web Agent in Policy Agent 2.2

The tasks that follow apply to Windows systems.

Generating a State File for a Web Agent Installation on Windows Systems

This section describes how to generate a state file for installing a web agent on Windows systems. The description that follows provides an option of performing this task through the GUI 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 Windows Systems

The following task describes how to generate a state file for a web agent installation on a Windows system.

1. **Change to the directory in which you unpacked the agent binaries.**

2. **Issue the command that applies as follows:**

   ❒ To use the GUI installation, issue the following command:

   ```
   java agent_WINNT_domino -saveState filename
   ```

   ❒ To use the command-line installation, issue the following command:

   ```
   java agent_WINNT_domino -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 "Installing the IBM Lotus Domino 7.0 Agent on Windows Systems" on page 38.**

   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 Windows Systems

This section describes how to use a state file for installing a web agent on Windows systems.

▼ To Install a Web Agent Using a State File on Windows 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 the web agent installation state file.

2 Issue the following command:

```
java agent_WINNT_domino -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 IBM Lotus Domino 7.0. 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 for the IBM Lotus Domino 7.0 Agent

This section includes various problems you might encounter with this agent on Solaris systems. The symptom of the problem is followed by possible causes and 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:

Launching installer... Sun Java(tm) System; Access Manager Policy Agent for IBM Lotus Domino 7.0 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>SUNWamdmn
        <compversion>2.2
                <uniquename>SUNWamdmn</uniquename>
                <vendor></vendor>
                <compinstance>1
                        <parent>Agent for Lotus Domino HTTP Server
                                <instance>1
                                        <version>2.2</version>
                                </instance>
                        </parent>
                <comptype>COMPONENT</comptype>
                <location>/opt/dm22</location>
                <dependent>
                                <compref>Agent for Lotus Domino HTTP Server
                                        <instance>1
                                                <version>2.2</version>
                                        </instance>
                                </compref>
                </dependent>
                         <data>
```

```
                                <key>pkgs
                                        <value>SUNWamdmn</value>
                                </key>
                        </data>
                </compinstance>
        </compversion>
</compid>
<compid>Agent for Lotus Domino HTTP Server
        <compversion>2.2
            <uniquename>Agent for Lotus Domino HTTP Server</uniquename>
               <vendor></vendor>
                 <compinstance>1
                       <parent>Sun Java(tm) System Access Manager Policy Agent
                             <instance>4
                                      <version>2.2</version>
                             </instance>
                       </parent>
                        <children>
                               <compref>SUNWamdmn
                                       <instance>1
                                         <version>2.2</version>
                                       </instance>
                              </compref>
                        </children>
                                        <comptype>FEATURE</comptype>
                                        <location>/opt/dm22</location>
                                        <dependent>
                               <compref>Sun Java(tm) System Access Manager Policy Agent
                                     <instance>4
                                           version>2.2</version>
                                     </instance>
                               </compref>
                                        </dependent>
                        <required>
                               <compref>SUNWamdmn
                        <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 Cause: 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 hosts.

Solaris Systems: Troubleshooting Symptom 4

Symptom: IBM Lotus Domino 7.0 server starts with the following error message:

```
Unable to load filter
```

Possible Cause: The DSAPI filter is configured incorrectly. Generally, if any path issue or associated library issue occurs while the DSAPI filter is being added, this error is generated.

Possible Solution: Ensure that the DSAPI filter has been configured with the correct information. For example, verify that the following path has been specified:

PolicyAgent-base`/SUNWam/agents/domino/lib/libamdomino6.so`

Solaris Systems: Troubleshooting Symptom 5

Symptom: The DSAPI filter is not functioning properly on a server instance.

Possible Causes:

- When the DSAPI filter was configured, the database selected was not correct.
- The DSAPI filter change may not be updated in the database

Possible Solutions:

- Ensure that the correct database was selected during configuration.
- Replicate the database from the IBM Lotus Domino 7.0 administration server.

Solaris Systems: Troubleshooting Symptom 6

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.

Solaris Systems: Troubleshooting Symptom 7

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 for the IBM Lotus Domino 7.0 Agent

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: IBM Lotus Domino 7.0 server starts with the following error message:

```
Unable to load filter
```

Possible Cause: The DSAPI filter is configured incorrectly. Generally, if any path issue or associated library issue occurs while the DSAPI filter is being added, this error is generated.

Possible Solutions:

- Ensure that the DSAPI filter has been configured with the correct information. For example, verify that the following path has been specified:
- Ensure that `LIBPATH` is set properly as explained in "AIX Systems: Setting `LIBPATH` to Include Libraries Specific to the IBM Lotus Domino 7.0 Agent" on page 59.

PolicyAgent-base`/agents/domino6/lib/libamdomino6.a`

AIX Systems: Troubleshooting Symptom 3

Symptom: The DSAPI filter is not functioning properly on a server instance.

Possible Causes:

- When the DSAPI filter was configured, the database selected was not correct.
- The DSAPI filter change may not be updated in the database

Possible Solutions:

- Ensure that the correct database was selected during configuration.
- Replicate the database from the IBM Lotus Domino 7.0 administration server.

AIX Systems: Troubleshooting Symptom 4

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 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
```

Windows Systems: Troubleshooting Symptoms for the IBM Lotus Domino 7.0 Agent

This section includes various problems you might encounter with this agent on Windows systems. The symptom of the problem is followed by possible causes and 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 IBM Lotus Domino 7.0

1. **Using the Lotus Domino Web console, remove the `amdomino6.dll` file from the DSAPI filter field.**
2. **Stop the Domino HTTP server.**
3. **Remove Agent for IBM Lotus Domino 7.0.**
   a. **In the Start menu, select Control Panel->Add/Remove programs**

b. **Select Sun Java System Access Manager Lotus Domino Agent**

c. **Click Remove**

4 **Remove entries from the product registry**

a. **Issue the following command in the command line:**

```
regedit
```

b. **Traverse to the following:**

```
HKEY_LOCAL_MACHINE
```

c. **Click Software**

d. **Click Sun Microsystems**

e. **Remove the following entry:**

Access Manager Lotus Domino Agent

5 **Remove the *PolicyAgent-base* directory from the server.**

where *PolicyAgent-base* represents the directory in which the web agent was originally installed.

6 **Remove the following entries from the `PATH` variable:**

- *PolicyAgent-base*`\bin`
- *PolicyAgent-base*`\domino\bin`

7 **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 Cause: 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: IBM Lotus Domino 7.0 server starts with the following error message:

```
Unable to load filter
```

Possible Cause: The DSAPI filter is configured incorrectly. Generally, if any path issue or associated library issue occurs while the DSAPI filter is being added, this error is generated.

Possible Solution: Ensure that the DSAPI filter has been configured with the correct information. For example, verify that the following path has been specified:

PolicyAgent-base`\\domino\\bin\\amdomino6.dll`

Windows Systems: Troubleshooting Symptom 4

Symptom: The DSAPI filter is not functioning properly on a server instance.

Possible Causes:

- When the DSAPI filter was configured, the database selected was not correct.
- The partitioned database was not updated.

Possible Solutions:

- Ensure that the correct database was selected during configuration.
- Replicate the database from the IBM Lotus Domino 7.0 administration server.

Windows Systems: Troubleshooting Symptom 5

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.

Windows Systems: Troubleshooting Symptom 6

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 Symptom 7

Symptom: When a user attempts to access a resource using a browser, access is denied.

Possible Cause: One or more properties in the web agent `AMAgent.properties` configuration file is set incorrectly. Specific properties, as specified in the following "Possible Solution" section, can cause access to resources to be denied.

Possible Solution: Verify that the values of the following properties are set correctly:

- `com.sun.am.naming.url`
- `com.sun.am.policy.am.login.url`
- `com.sun.am.policy.am.username`
- `com.sun.am.policy.am.password`

Windows Systems: Troubleshooting Symptom 8

Symptom: After the agent is installed, the web server fails to start.

Possible Cause: Libraries that agents depend on for Windows systems are missing. Ensuring that the libraries `msvcp70.dll` and `msvcr70.dll` are available is a pre-installation step in this guide. If the libraries were not properly added, the web server might not start.

Possible Solution: Obtain the appropriate libraries as described in "Preparing to Install the IBM Lotus Domino 7.0 Agent on Windows Systems" on page 37.

Linux Systems: Troubleshooting Symptoms for the IBM Lotus Domino 7.0 Agent

This section includes a problem you might to encounter with this agent on Linux systems. The symptom of the problem is followed by possible causes and solutions.

Linux Systems: Troubleshooting Symptom 1

Symptom: IBM Lotus Domino 7.0 server starts with the following error message:

```
Unable to load filter
```

Possible Cause: The DSAPI filter is configured incorrectly. Generally, if any path issue or associated library issue occurs while the DSAPI filter is being added, this error is generated.

Possible Solution: Ensure that the DSAPI filter has been configured with the correct information. For example, verify that the following path has been specified:

PolicyAgent-base`/agents/domino6/lib/libamdomino6.so`

Linux Systems: Troubleshooting Symptom 2

Symptom: The DSAPI filter is not functioning properly on a server instance.

Possible Causes:

- When the DSAPI filter was configured, the database selected was not correct.
- The partitioned database was not updated.

Possible Solutions:

- Ensure that the correct database was selected during configuration.
- Replicate the database from the IBM Lotus Domino 7.0 administration server.

Linux Systems: Troubleshooting Symptom 3

Symptom: The agent goes into an infinite loop.

Possible Cause: The value 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.

Linux Systems: Troubleshooting Symptom 4

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 75.

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
com.sun.am.policy.agents.config. iis6.basicAuthentication.username	NEW PROPERTY
com.sun.am.policy.agents.config. iis6.basicAuthentication.password	NEW PROPERTY
com.sun.am.policy.agents.config. iis6.basicAuthentication.logFile	NEW PROPERTY
com.sun.am.policy.agents.config. domino.check_name_database	NEW PROPERTY
com.sun.am.policy.agents.config. domino.ltpa.enable	NEW PROPERTY
com.sun.am.policy.agents.config. domino.ltpa.cookie_name	NEW PROPERTY
com.sun.am.policy.agents.config. domino.ltpa.config_name	NEW PROPERTY
com.sun.am.policy.agents.config. domino.ltpa.org_name	NEW PROPERTY
com.sun.am.policy.agents.config. domino.checkNameDatabase	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

D

E

F

G

H

S

T

U

V

W