

J2EE Policy Agents Guide

Sun™ ONE Identity Server Policy Agents

Version 2.1

816-6884-10
April 2005

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

Copyright 2005 Sun Microsystems, Inc. All rights reserved.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun ONE, iPlanet, and all Sun, Java, and Sun ONE based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Netscape is a trademark or registered trademark of Netscape Communications Corporation in the United States and other countries.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of Sun Microsystems, Inc. and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2005 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, le logo Sun, Java, Solaris, Sun ONE, et iPlanet sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et d'autres pays.

UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Netscape est une marque de Netscape Communications Corporation aux Etats-Unis et dans d'autres pays.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de Sun Microsystems, Inc. et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE "EN L'ÉTAT", ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

About This Guide	9
What You Are Expected to Know	9
Sun ONE Identity Server Documentation Set	10
Documentation Conventions Used in This Manual	10
Typographic Conventions	11
Terminology	11
Related Third-Party Web Site References	11
Related Information	12
Chapter 1 Read This First	13
Uses of J2EE Policy Agents	14
General Usage Examples	14
Specialized Agent Usage Examples	17
How J2EE Policy Agents Work	20
PeopleSoft 8.3/8.4/8.8 Agent Architecture	21
Supported Servers	24
What's New in J2EE Policy Agents	26
URL Policy Support	26
Web-Tier Declarative Security Support	26
Other Features	26
Differences Between J2EE Policy Agents and Web Policy Agents	27
Differences in Protected Resources	27
Default Scope of Protection	28
Modes of Operation	29
Different Configuration Properties	29

Chapter 2 Installing the Agent	31
Pre-Installation Tasks	31
Common Tasks	32
Agent for Sun ONE Application Server 7.0	33
Agent for BEA WebLogic 6.1 SP2	33
Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1	33
Agent for IBM WebSphere 5.0/5.1	34
Agent for PeopleSoft 8.3/8.4/8.8	34
Agent for Apache Tomcat Server 4.1.27	39
Agent for Oracle 9iAS R2 and Oracle 10g	40
Agent for SAP Enterprise Portal 6.0 SP2	41
Agent for Macromedia JRun 4	42
Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1	43
Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal	45
Agent for Sun Java System Application Server 8.1	45
Launching the Installation Program	46
Launching the Installation Program on Solaris, HP-UX, AIX, and Linux	46
Launching the Installation Program on Windows	49
Using the Installation Program	51
Using the GUI Installation Program	51
Using the Command-Line Installation Program	83
Post-Installation Tasks	107
Agent for Sun ONE Application Server 7.0	108
Agent for BEA WebLogic Server 6.1 SP2	109
Agent for IBM WebSphere Application Server 5.0/5.1	114
Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1	116
Agent for PeopleSoft 8.3/8.4/8.8	120
Agent for Apache Tomcat Server 4.1.27	128
Agent for Oracle 9iAS R2 and Oracle 10g	130
Agent for SAP Enterprise Portal 6.0 SP2	134
Agent for Macromedia JRun 4	136
Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1	139
Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal	143
Agent for Sun Java System Application Server 8.1	149
Common Tasks	151
Customizing the Agent Installation	152
Chapter 3 Agent Configuration	153
General Notes on the Agent Configuration File	154
Hot-Swap Mechanism	154
List Constructs in the Configuration File	154
Map Constructs in the Configuration File	155
Agent Filter Modes	156

Agent Filter Mode - NONE	157
Agent Filter Mode - SSO_ONLY	157
Agent Filter Mode - J2EE_POLICY	158
Agent Filter Mode - URL_POLICY	158
Agent Filter Mode - ALL	159
Disabling the Agent Realm	159
Agent for Sun ONE Application Server 7.0	159
Agent for BEA WebLogic Server 6.1 SP2	160
Agent for IBM WebSphere Application Server 5.0/5.1	160
Agent for Oracle 9iAS and Oracle 10g	161
Agent for Sun Java System Application Server 8.1	162
Hot-Swap Configuration	162
Enabling Web-Tier Declarative Security	163
Web-Tier Security Details	168
Customizing Agent Response for Form Login	169
Enabling Failover	170
Login Attempt Limit	171
Redirect Attempt Limit	172
Not-Enforced List	172
Inverting the Not-Enforced List	173
Enabling LDAP Attributes	174
LDAP Attributes as HTTP Headers	174
LDAP Attributes as Request Attributes	175
LDAP Attributes as Cookies	175
Configuring FQDN Handling	176
Using Cookie Reset Functionality	177
Enabling Port Check Functionality	178
AMAgent.properties Reference	179
Agent Filter Mode	182
Configuration Reload Interval	184
Language Code	185
Country Code	186
Registered Module List	187
Login URL	187
Counter Cookie Name	188
Login Attempt Limit	189
URL Decode SSO Token Flag	190
Goto Parameter Name	191
Session Binding Flag	192
Not-Enforced List	193
Not-Enforced List Inversion Flag	194
Not-Enforced-List Cache Enable Flag	195
Not-Enforced-List Cache Size	196

Not-Enforced-List Cache Expiration Time	197
Access Denied URI	199
LDAP Date Header Attribute Format String	200
LDAP Attribute Map	201
LDAP Attribute Fetch Mode	202
LDAP Attribute Cookie Separator Character	204
LDAP Attribute Cookie Encode	205
FQDN Default	206
FQDN Map	207
J2EE Authentication Handler	208
J2EE Logout Handler	209
Login Form List	210
Form Login Use Internal Flag	212
Form Login Content File Name	213
Preserve Referer for Form Login Flag	214
Default Referer Map for Form Login	216
Cookie Reset Enable Flag	217
Cookie Reset List	218
Cookie Reset Domain Map	219
Cookie Reset Path Map	220
Audit Log Level	221
Redirect Counter Cookie Name	223
Redirect Attempt Limit	223
Legacy User Agent Support Flag	224
Legacy User Agent List	225
Legacy User-Agent Intermediate Redirect URI	227
Port Check Enable Flag	228
Port Check Map	229
Port Check Content File Name	230
Fetch All Operation Flag	231
People Container Level	232
Organization DN	233
Audit Log Disposition	234
Audit Log Local File Name	235
Audit Log Local File Rotate Flag	236
Audit Log Local File Rotation Size	237
Audit Log Remote File Name	238
Bypass Principal List	239
PeopleSoft User Mapping	240
User Attribute Containing PeopleSoft User	241
Validate SSO Token in PeopleCode flag	242
URL Decode SSO Token Flag	243
Authentication Module	243

Goto URL	244
Display Resource Root	245
Default Display Resource File Name	246
Display Resource Map	247
CDSSO Enable Flag	248
CDC Servlet URL	249
CDSSO Intermediate Redirect URI	250
CDSSO Request Cookie Name	251
CDSSO Liberty Assertion Validity Clock Skew Factor	252
Login Form Error List	253
Agent Operation Mode Map	255
Enable Filtered Roles	257
SAP User Mapping	258
User Attribute Containing SAP User-ID	259
Logon Frontend Tracking Cookie Name	260
Goto URL	261
Error Content File Name	261
User Mapping	262
User Mapping Mode	264
User Attribute Containing Agent User	265
Logout Parameter Map	266
Logout URI Map	268
Application Authentication Handler Map	269
Verification Handler Map	270
Global Verification Handler	271
HTTP Request Body Introspect Flag	272
Application Local Logout Handler Map	273
Chapter 4 Tools and APIs	275
Agent Tools	275
Using Tools to Encrypt Strings	276
Configuring the Agent for an Application Server Instance	277
Unconfiguring the Agent for an Application Server Instance	290
Agent APIs	299
Class AmFilterManager	299
Class AmSSOCache	300
Chapter 5 Uninstalling the Agent	301
Pre-Uninstallation Tasks	301
Agent for Sun ONE Application Server 7.0	302
Agent for BEA WebLogic Server 6.1 SP2	302
Agent for BEA WebLogic Server 7.0 SP2/8.1	304

Agent for IBM WebSphere 5.0/5.1	304
Agent for PeopleSoft 8.3/8.4/8.8	305
Agents for Oracle 9iAS R2 and Oracle 10g	305
Agent for Tomcat Server 4.1.27	305
Agent for SAP Enterprise Portal 6.0 SP2	305
Agent for Macromedia JRun 4	305
Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1	306
Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal	307
Agent for Sun Java System Application Server 8.1	308
Launching the Uninstallation Program	308
Launching the Uninstallation Program on Solaris, HP-UX, AIX and Linux	308
Launching the Uninstallation Program on Windows 2000	311
Using the Uninstallation Program	312
Using the GUI Uninstallation Program	312
Using the Command Line Uninstallation Program	318
Post-Uninstallation Tasks	324
Agent for SAP Enterprise Portal 6.0SP2	325
Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1	326
Chapter 6 Securing Identity Manager Software With a Policy Agent	327
Deployment Facts and Considerations	327
The Policy Agent Installation	328
Configuring the Agent	328
Configuring the Identity Manager Software	329
Adding the Resource Adapter	329
Configuring the Resource Instance	330
Creating a Login Module Group	332
Assigning the Login Module Group to Login Applications	333
Provisioning Users Using Identity Manager Software	334
Testing SSO Integration With Identity Server	336
Protecting Identity Manager Using URL Policies	337
How to Troubleshoot the Agent Configuration	338
Appendix A Silent Installation	339
About Silent Installation/Uninstallation	339
Generating a State File for Installation	339
Using the State File for Silent Installation	340
Generating a State File for Uninstallation	341
Using the State File for Silent Uninstallation	342
Appendix B Sample Application Scenario	345
Standard Deployment Descriptors	345

web.xml	346
ejb-jar.xml	349
application.xml	350
Assembly Descriptors	350
Sun ONE Application Server 7.0	351
BEA WebLogic 6.1 SP2	352
IBM WebSphere Application Server 5.0/5.1	354
BEA WebLogic Server 7.0 SP2 and BEA WebLogic Server 8.1	357
Apache Tomcat Server 4.1.27	359
Macromedia JRun 4	360
Oracle 9iAS R2 and Oracle 10g	360
Sun Java System Application Server 8.1	362
Appendix C Troubleshooting the Agent Deployment	365
Index	377

About This Guide

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

This preface contains the following sections:

- [What You Are Expected to Know](#)
- [Sun ONE Identity Server Documentation Set](#)
- [Documentation Conventions Used in This Manual](#)
- [Related Information](#)

What You Are Expected to Know

This book is considered to be an auxiliary manual in the documentation series provided with Sun ONE Identity Server 6.0 SP1. It's essential that you understand directory technologies and have some experience with Java, J2EE and XML programming languages. You will get the most out of this guide if you are familiar with directory servers and Lightweight Directory Access Protocol (LDAP). Particularly, you should be familiar with Sun ONE Directory Server and the documentation provided with that product.

This guide is intended for use by IT professionals who manage access to their network through Sun ONE servers and services. The functionality contained in Sun ONE Identity Server Policy Agents allow you to protect hosted J2EE applications.

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

Sun ONE Identity Server Documentation Set

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

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

NOTE

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

Documentation Conventions Used in This Manual

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

Typographic Conventions

This book uses the following typographic conventions:

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

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

Terminology

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

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

Related Third-Party Web Site References

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

Related Information

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

Download Center

Links to download any of Sun's Sun ONE/iPlanet software are at

<http://www.sun.com/software/download/>

Sun ONE Technical Support

Technical Support can be contacted through

<http://www.sun.com/service/support/software/iplanet/index.html>

Professional Services Information

Professional Service can be contacted through

<http://www.sun.com/service/sunps/iplanet/>

Sun Enterprise Services for Solaris Patches And Support

Solaris patches and support can be obtained through

<http://www.sun.com/service/>

Developer Information

Information on Sun ONE Identity Server, LDAP, the Sun ONE Directory Server, and associated technologies can also be found at

<http://developer.iplanet.com/tech/directory/>

Read This First

Sun™ ONE Identity Server Policy Agents, version 2.1, comprise J2EE Policy Agents and Web Policy Agents. J2EE Policy Agents enable application servers to enforce authentication and authorization using Sun™ ONE Identity Server services. While ensuring secure client access to the hosted J2EE applications, the policy agents enforce J2EE security policies defined in the deployed application's deployment descriptors as well as URL policies defined in Sun ONE Identity Server.

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

This chapter provides a brief overview of J2EE Policy Agents. Topics in this chapter include:

- [Uses of J2EE Policy Agents](#)
- [How J2EE Policy Agents Work](#)
- [Supported Servers](#)
- [What's New in J2EE Policy Agents](#)

Uses of J2EE Policy Agents

The J2EE policy agent may be installed for protecting a variety of hosted J2EE applications, which may require a varying set of security policy implementation. The security infrastructure of J2EE provides declarative as well as programmatic security that are platform-independent and are supported by all the J2EE-compliant application servers. For details on how to use J2EE platform's declarative as well as programmatic security, refer to J2EE documentation, which can be found at <http://java.sun.com/j2ee>.

The agent helps enable role-to-principal mapping for protected J2EE applications with Sun ONE Identity Server principals. Thus at runtime, when a J2EE policy is evaluated, it is done against the information available in Sun ONE Identity Server. Using this functionality, administrators may configure their hosted J2EE applications to be protected by the agent, which provides real security services and also other key features such as single sign-on.

Apart from enabling the J2EE security for hosted applications, the J2EE policy agents also provide complete support for Sun ONE Identity Server-based URL policies for enforcing access control over web resources hosted in the application server's web container.

The following two sections provide examples of how the J2EE policy agents can be put to use:

- [General Usage Examples](#)
- [Specialized Agent Usage Examples](#)

The general usage examples provide usage scenarios on how most of the J2EE agents can be used to enforce access control over protected resources in J2EE application servers. The specialized agents, on the other hand, are specific to application platforms such as PeopleSoft and SAP servers which allow single sign-on and policy enforcement over such specialized resources.

General Usage Examples

An Online Auction Application

Consider a web-based application that facilitates the auction of various merchandise between interested parties. A simple implementation for such an application will require the users to be in one of three abstract roles, namely Buyer, Seller or Administrator. Buyers in this application will have access to web pages

that display the listed auction items, whereas the Sellers may have access to web pages that allow them to list their merchandise for new auctions. The Administrators may have access to yet another set of web pages that allow them to finalize or cancel existing auctions in whatever state they may be in.

Using the deployment descriptors, the application developer can express this intent by protecting such components using abstract security role names. These abstract role names in turn can be mapped to real principals in Sun ONE Identity Server using the policy agent. For example, the role Buyer may be mapped to a Sun ONE Identity Server role called Employee, the role Seller to a Sun ONE Identity Server role called Vendor, and the role Administrator to a Sun ONE Identity Server role called Admin.

The abstract role names used by the application developer may be used to protect the necessary web pages and any specialized Enterprise JavaBeans (EJB) components from unauthorized access by using declarative as well as programmatic security. Once this application is deployed and configured, the agent will ensure that only the authorized personnel get access to these protected resources. For example, access to the pages meant for Sellers to list their merchandise for auctions will be granted to the user Shariff only if this user belongs to the Sun ONE Identity Server role called Vendor. Similarly, users Navaneeth and Vasanth may place bids on this listed item only if they belong to the role called Buyer. Once the auction period expires, the auction can be finalized by the user Krishnendu only if he is in the role called Admin.

A Web-Based Commerce Application

A web-based commerce application may have a variety of specialized EJB components that offer a spectrum of services to the clients. For instance, there could be a specialized component that enables to create purchase orders. Similarly, there could be a specialized component that allows to approve a purchase order. While such components provide the basic business services for the application to function, the very nature of tasks that they accomplish requires a security policy to enforce appropriate use of such services.

Using the deployment descriptors, the application vendor or developer can express this intent by protecting such components using abstract security role names. For example, there can be a role called Buyer, which protects the component that allows to create a purchase order. Similarly, there can be a role called Approver, which protects the component that enables to approve a purchase order. While these roles convey the intent of the application vendor or developer to enforce such security policies, they will not be useful unless these abstract role names are mapped to real life principals such as actual users or actual roles that reside in Sun ONE Identity Server.

The agent provides the ability to the container to enforce such a runtime linkage of abstract security roles to real life principals. Once the agent is installed and configured, the application security roles can be mapped to real principals. For example, the role Buyer can be mapped to a Sun ONE Identity Server role called Staff, and the role Approver can be mapped to a Sun ONE Identity Server role called Manager. Thus when a user Arvind tries to access the application's protected resources to create a purchase order, the agent will allow this access only if this user is a member of the mapped role Staff. Similarly, a user Maya may wish to approve this purchase order, which will be allowed by the agent only if this user is a member of the mapped role Manager.

A Content-Based Web Application

A content-based web application can offer pay per-view services. The application may be partitioned into two domains: the public domain that is accessible to anonymous users, and the private domain that is accessible only to the subscribers of this particular service. Further, the protected domain of this application can also be subject to strict conditions based on how the user has authenticated, the time of the day, IP address-based conditions and so on.

Using Sun ONE Identity Server-based URL policies for web resources, the Administrator can specify such complex policies for the application resources, which will be evaluated by the policy agent in order to ensure that access to these resources is granted only when all conditions are satisfied. The Administrator can set policies that govern access to these resources at any level of granularity - such as that for a single user or for an entire organization.

For example, one such policy may govern access to certain resources in such a manner that the user must belong to a particular LDAP Group called Customer and that the time of the day be between 9:00 am and 5:00 pm. Thus, if a user Rajeev were to access this resource, the agent will allow this access only if this user is a member of the LDAP Group Customer, and that the time of the day is between 9:00 am and 5:00 pm.

Specialized Agent Usage Examples

Policy Agent for PeopleSoft Applications

Consider an enterprise where PeopleSoft human resources and payroll applications are deployed along with other web applications. The enterprise may want to achieve single sign-on across different applications including PeopleSoft applications. This means, that on successfully logging in with sufficient credentials, the user should get seamless access to all the applications without having to re-authenticate as long as he/she is authorized to access the resource.

For example, if a user Pradeep is trying to access PeopleSoft application, he will be redirected to Sun ONE Identity Server for authentication if he is not already authenticated. On successful login, Pradeep will be allowed to access his payroll information as well as personal information from the PeopleSoft human resource application.

Policy Agent for SAP Enterprise Portal

The J2EE Policy Agent allows seamless integration of SAP Enterprise Portal and its hosted portal applications and content with Sun ONE Identity Server and allows the administrators to enforce stricter access control policies, where necessary. It also allows for a very flexible mode of mapping Sun ONE Identity Server users to users as they exist in the SAP back-end system or Enterprise Portal user base. One way to map the Sun ONE Identity Server users with SAP Enterprise Portal users is to configure the agent to use the same user ID across both the systems. For example, if the user Rajesh signs on to Sun ONE Identity Server using a user ID `rajesh`, the agent when configured to map the same user IDs will log him into the SAP Enterprise Portal as `rajesh`.

In situations where it is not possible to map the same user ID across Identity Server and SAP Enterprise Portal, the agent may be configured to use a mapped user ID from the LDAP attribute of the Identity Server user. For instance, if a user Sushma has the value `sushmac` stored in a particular profile attribute, which the agent has been configured to read, then she will be logged into Enterprise Portal as user `sushmac` by the agent when she accesses the Enterprise Portal resources.

The agent also allows a third mode of mapping users of Identity Server to users in SAP Enterprise Portal, which is by using the value of a specified HTTP Header field. Thus if a user Amlan requests access to a protected Enterprise Portal resource, and this request carries with it a value `amlanc` in an HTTP header field that the agent has been configured to read, then the agent will log him into the SAP Enterprise Portal as user `amlanc`.

The agent for SAP Enterprise Portal can also be used to enforce URL Policies defined in Identity Server's console to control access to the protected Enterprise Portal resources. For example the administrator may define a set of URL Policies in the Identity Server such that these policies will allow access to a particular URL resource in Enterprise Portal to the user Frances and not to the user Deepak. In this case, regardless of the access control set up within Enterprise Portal for the resource in question, the agent will only allow access to the user Frances for that particular resource and not for the user Deepak. Furthermore, the administrator can also set up URL policies with various conditions and authentication levels or scheme requirements for such resources. As an example, the administrator may create a policy for user Bill such that the access to a particular Enterprise Portal resource is restricted to a particular time of the day only. In this case, the user Bill will be allowed access to the given resource only during the allowed interval and will be denied access at all other times.

Policy Agent for SAP Enterprise Portal and Web Application Server

The Policy Agent for SAP Enterprise Portal and Web Application Server is similar to the Policy Agent for SAP Enterprise Portal in that they both:

- allow seamless integration with Sun ONE Identity Server
- facilitate single sign-on between their deployed applications and the applications and servers protected by Sun ONE Identity Server.
- unlike J2EE Policy Agents for various other application servers, do not facilitate role-to-principal mapping using the Identity Server role definitions. Instead they simply enable the establishment of Identity Server principals within the hosted applications on SAP Web Application Server.

However, Policy Agent for SAP Enterprise Portal and Web Application Server is different from Policy Agent for SAP Enterprise Portal since between the two only Policy Agent for SAP Enterprise Portal and Web Application Server allows the establishment of Identity Server principals within the applications hosted on the underlying SAP Web Application Server.

For example, when a user attempts to access an application that is hosted on SAP Web Application Server and is protected by Policy Agent for SAP Enterprise Portal and Web Application Server, the policy agent ensures user authentication and enforces any applicable URL policies defined within Identity Server. This authentication and enforcement by the agent establishes the principal within the SAP Web Application Server for the protected application. And yet, to ensure complete protection of hosted resources, the administrator of SAP Web Application Server needs to define the necessary protection domains, resource protection policies and constraints within the SAP Administrator as well as the deployed application.

Policy Agent for BEA WebLogic 8.1 Server/Portal

This J2EE agent was developed with the objective of protecting J2EE web applications deployed on BEA WebLogic 8.1 SP2/SP3 Server (BEA WebLogic SP2 Server or BEA WebLogic SP3 Server) and on BEA WebLogic 8.1 SP2/SP3 Portal (BEA WebLogic SP2 Portal or BEA WebLogic SP3 Portal). In this document, the agent is commonly referred to as WebLogic 8.1 Server/Portal Agent.

This particular section describes a few use cases for this agent when installed specifically on WebLogic 8.1 Portal. Users are allowed to access a WebLogic 8.1 Portal application if their user profile is present in both Identity Server and the WebLogic 8.1 Portal repository. Administrators must ensure that users who can log into the WebLogic Portal are also present in Identity Server. Users who are not present in either of the two repositories should be denied access to any resource they try to access.

Note that after installing and configuring the agent for a portal application, the content management of the portal application is still performed by the WebLogic 8.1 Portal framework. The J2EE agent simply achieves single sign on by authenticating the user with Identity Server and by establishing the user principal in the portal container. The agent also supports synchronization of the Identity Server and WebLogic 8.1 Portal sessions through a logout event trapping feature.

This agent allows for a very flexible mode of mapping Sun ONE Identity Server users to users as they exist in the WebLogic 8.1 Portal repository. One way to map the Sun ONE Identity Server users with WebLogic 8.1 Portal users is to configure the agent to use the same user ID across both systems. For example, if the user JinWon signs on to Sun ONE Identity Server using a user ID `jinwon`, the agent—when configured to map the same user ID—logs him into the WebLogic 8.1 Portal as `jinwon`.

In situations where it is not possible to map the same user ID across Identity Server and the WebLogic 8.1 Portal, the agent can be configured to use a mapped user ID from the LDAP attribute of the Identity Server user. For instance, if a user Gina has the value `Ginak` stored in a particular profile attribute that the agent has been configured to read then she will be logged into WebLogic 8.1 Portal as user `Ginak` by the agent when she accesses the WebLogic 8.1 Portal resources.

The agent also allows a third mode of mapping users of Identity Server to users in the WebLogic 8.1 Portal, by using the value of a specified HTTP Header field. Thus if a user BethAnn requests access to a protected WebLogic 8.1 Portal resource, and this request carries with it a value `bethann` in an HTTP header field that the agent has been configured to read, then the agent will log her into the WebLogic 8.1 Portal as user `bethann`.

The agent for the WebLogic 8.1 Portal can also be used to enforce URL Policies defined in Identity Server's console to control access to the protected WebLogic 8.1 Portal. For example the administrator can define a set of URL Policies in the Identity Server such that these policies allow access to a particular URL resource in the WebLogic 8.1 Portal to the user Rachel and not to the user David. In this case, regardless of the access control set up within WebLogic 8.1 Portal for the resource in question, the agent only allows access to Rachel for that particular resource and not for David. Furthermore, the administrator can also set up URL policies with various conditions and authentication levels or scheme requirements for such resources.

How J2EE Policy Agents Work

J2EE Policy Agents contain two main components that together affect the operation of the application server as well as the behavior of the protected applications. These components are as follows:

- **Agent Realm**

The agent realm is installed as an application server-specific platform component, which provides the ability to the application server to interact with principals stored in Sun ONE Identity Server. This component needs to be configured correctly in order that the agent can be used to enforce J2EE Security policies for the protected application. This component is needed for enforcing J2EE policies for the protected application.

NOTE Because the agent realm component is installed as an application server-specific component, it may be named differently for different agents. For example, it is called custom registry for IBM WebSphere 5.0/5.1 agent, and agent authentication provider for BEA WebLogic 7.0 SP2 and 8.1 agents.

- **Agent Filter**

The agent filter is installed within the protected application and facilitates the enforcement of the security policies governing the access to all resources within the protected application. Every application that has to be protected by the agent must have its deployment descriptors changed to reflect that it is configured to use the agent filter. Applications that do not have this setting will not be protected by the agent and may malfunction or become unusable if deployed on an application server where the agent realm is installed.

NOTE For application servers that do not support Servlet Filters as defined in the Servlet 2.3 specification, the agent filter may be named and packaged differently. For example, the agent filter is referred to as the authfilter for the WebLogic front-end server used with PeopleSoft 8.3 server.

Together, the agent realm and agent filter work in tandem with Sun ONE Identity Server and enforce authentication and authorization for clients trying to access protected J2EE applications by enforcing the J2EE security policies as well as Sun ONE Identity Server-based URL policies where applicable.

The agent provides a fully configured and ready-to-use client installation of Sun ONE Identity Server SDK for the application server. This SDK offers a rich set of APIs supported by Sun ONE Identity Server that can be used to create security-aware applications that are tailored to work in the security framework offered by Sun ONE Identity Server. For more information on how to use Sun Java System Identity Server SDK, please refer to Sun Java System Identity Server Programmer's Guide.

PeopleSoft 8.3/8.4/8.8 Agent Architecture

The policy agent for PeopleSoft 8.3, 8.4 and 8.8 protects web access to PeopleSoft applications using Sun ONE Identity Server security features such as authentication, single sign-on and policy.

PeopleSoft applications are web-enabled through PeopleSoft Internet Architecture (PIA) that involves deploying PeopleSoft servlets on a web/application server (WebLogic) tier, which fronts the PeopleSoft proprietary application server. The policy agent can either be installed on the web/application server or on a web container that acts as proxy to the PeopleSoft web/application server.

The policy agent for PeopleSoft is based on the Sign-on PeopleCode provided by PeopleSoft. For more information on this infrastructure, see Chapter 7 of the Security PeopleBook that is part of the PeopleSoft documentation.

The policy agent code on the user's web/application server (either the proxy or the PeopleSoft-provided web/application server) works like a regular Sun ONE Identity Server web agent. It intercepts all the user-requests and is responsible for authenticating and authorizing web access to the PeopleSoft application based on the authentication schemes and policies defined in Sun ONE Identity Server.

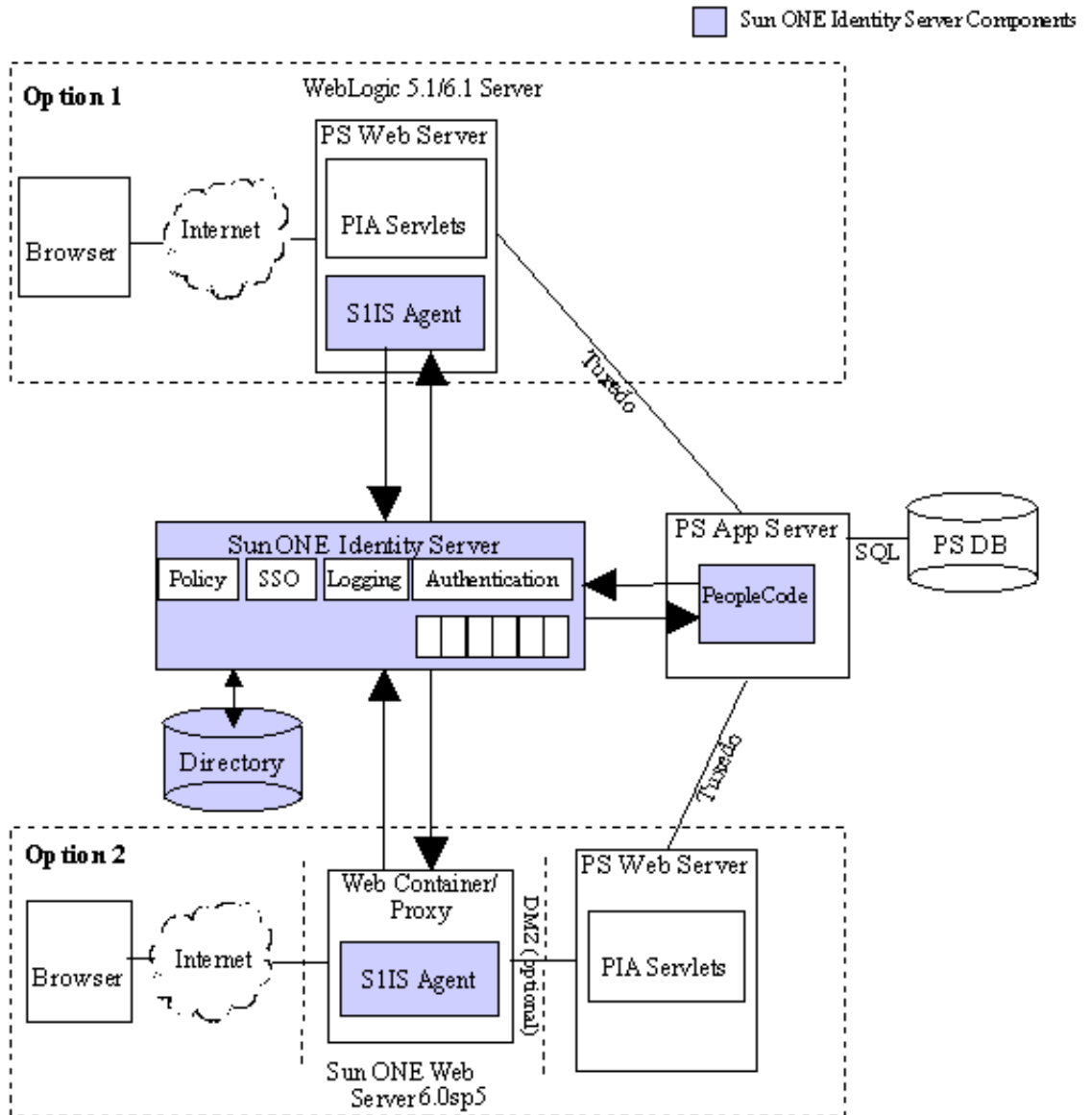
The PeopleSoft configuration, required for the agent to work, involves the following three steps:

1. Creating a default user in PeopleSoft.
2. Configuring the web server servlets to pass on the default user to the application server.
3. Configuring the application server to delegate the process of determining the user logged in to a special PeopleCode (PeopleSoft's proprietary language) provided by Sun ONE Identity Server Policy Agent.

The rest of PeopleSoft configuration, specifically the existing PeopleSoft user repository, user profiles and PeopleSoft roles based access control do not need any changes, and work exactly like before.

The current implementation of identifying the user to PeopleSoft assumes that Sun ONE Identity Server and PeopleSoft separately maintain users, and a pre-determined LDAP attribute in the Sun ONE Identity Server user's entry contains the PeopleSoft userid. User synchronization between the two user repositories is beyond the scope of Sun ONE Identity Server or the Sun ONE Identity Server Policy Agent. While continuing to support this form of mapping, future implementations will provide support for installations that have a single user LDAP repository shared between Sun ONE Identity Server and PeopleSoft.

Figure 1-1 PeopleSoft Agent Architecture



Supported Servers

J2EE Policy Agents, versions 2.1 and 2.1.1 are available for the following servers. These versions of the agents work with Sun ONE Identity Server 6.0 SP1, 6.1 and 6.2. The agent supported on Solaris 8 is generally supported on Solaris 9 also and vice versa.

NOTE J2EE Policy Agents support only specific versions of Oracle 9iAS and Oracle 10g application servers as listed in the following table. Hence, before installing the agent for Oracle 9iAS or Oracle 10g, be sure to check the exact version of the application server from the property named `Version` in the file `ORACLE_HOME/config/ias.properties`.

Table 1-1 Servers Supported by J2EE Policy Agents (1 of 3)

Server	Platform	Agent Version Available
Sun ONE Application Server 7.0	Solaris 8	2.1, 2.1.1
	Solaris 9	2.1, 2.1.1
	Microsoft Windows 2000	2.1, 2.1.1
	Red Hat Advanced Server 2.1	2.1, 2.1.1
BEA WebLogic Server 6.1 SP2	Solaris 8	2.1, 2.1.1
	Solaris 9	2.1, 2.1.1
	HP-UX 11	2.1, 2.1.1
	Microsoft Windows 2000	2.1, 2.1.1
BEA WebLogic Server 7.0 SP2	Solaris 8	2.1
	Solaris 9	2.1
	Red Hat Advanced Server 2.1	2.1
	HP-UX 11.11	2.1
IBM WebSphere Application Server 5.0	Solaris 8	2.1, 2.1.1
	Microsoft Windows 2000	2.1, 2.1.1
	AIX 5.1	2.1
	AIX 5.2	2.1.1
	Red Hat Advanced Server 2.1	2.1.1
PeopleSoft 8.3/8.4/8.8	Solaris 8	2.1

Table 1-1 Servers Supported by J2EE Policy Agents (2 of 3)

Server	Platform	Agent Version Available
	HP-UX 11	2.1
BEA WebLogic Server 8.1	Solaris 8	2.1
	Solaris 9	2.1
	Red Hat Advanced Server 2.1	2.1
	HP-UX 11.11	2.1
Apache Tomcat Server 4.1.27	Solaris 8	2.1
	Solaris 9	2.1
	Red Hat Advanced Server 2.1	2.1
Oracle 9i Application Server Release 2 (version 9.0.3)	Solaris 8	2.1
SAP Enterprise Portal 6.0 SP2	Solaris 8	2.1
	Solaris 9	2.1
	AIX 5.2	2.1
Oracle Application Server 10g (versions 9.0.3 and 9.0.4)	Solaris 8	2.1
Macromedia JRun 4 Application Server	Solaris 8	2.1, 2.1.1
	Solaris 9	2.1, 2.1.1
	Red Hat Enterprise Linux 3.0	2.1, 2.1.1
SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1	Microsoft Windows Server 2003 Enterprise Edition	2.1
BEA WebLogic 8.1 SP2 Server/Portal	Solaris 8	2.1
	Solaris 9	2.1
BEA WebLogic 8.1 SP3 Server/Portal	Red Hat Enterprise Linux AS 3.0	2.1
	Solaris 8	2.1
	Solaris 9	2.1
IBM WebSphere Application Server 5.1	Solaris 8	2.1.1
Sun Java™ System Application Server 8.1	Solaris 8	2.1
	Solaris 9	2.1
	Solaris 10	2.1

Table 1-1 Servers Supported by J2EE Policy Agents *(3 of 3)*

Server	Platform	Agent Version Available
	Solaris 9 x86	2.1
	Red Hat Advanced Server 2.1	2.1

What's New in J2EE Policy Agents

J2EE Policy Agents support a variety of new features. Some of the features require application configuration as well as agent configuration in order to be used correctly. Make sure that you read the chapters on installation and agent configuration to ensure the proper usage of these features.

URL Policy Support

This release of J2EE Policy Agents provides complete support for Sun ONE Identity Server-based URL policies that can be used to protect web-tier resources such as URLs that may point to servlets, JSPs, HTML files, images and other types of web-tier resources.

Web-Tier Declarative Security Support

The web-tier declarative security implies the use of deployment descriptors to control access for various web-tier resources such as servlets, JSPs, HTML files, images and so on. The policy agents help enforce such access control in the application server by establishing linkage between abstract role names used in the deployment descriptors of such web applications with real principals as available in the Sun ONE Identity Server.

Other Features

This release of J2EE Policy Agents also has a host of other new features that can be used to better secure your deployed applications as well as customize the agent environment for best use in your deployment. Complete details of these features and how best to use them are available in the following chapters. A brief list of these features is as follows:

- Support for hot-swap configuration
- Ability to set LDAP Attributes as Cookies
- Ability to set LDAP Attributes as Request Attributes
- Remote logging capability
- Ability to reset application cookies on unauthorized access to system resources
- Support for various modes of operation of the agent filter
- Support for Login and Naming Service Failover
- Support for inversion of Not-Enforced List
- Support for single point redirection loop protection

Differences Between J2EE Policy Agents and Web Policy Agents

If you have already used or are planning to use both J2EE Policy Agents and Web Policy Agents, you need to familiarize yourself with certain key differences between these two types of agents. Typically, the Web Policy Agents are used to protect resources hosted on web servers or enforce Single Sign-On with systems that use web servers as front-end in an environment secured by an Identity Server deployment. On the other hand, the J2EE Policy Agents are used to protect resources hosted on J2EE Application Servers or enforce Single Sign-On with systems that use J2EE Application Servers as their underlying infrastructure in an environment secured by an Identity Server deployment.

While the primary purpose of both these types of agents is to enforce authentication and authorization before a user can access a protected resource, they differ greatly in the kind of resources that they can protect, in the way they enforce such policy decisions, and in the way they can be configured to provide support for features that are solely applicable to the kind of servers they protect.

Differences in Protected Resources

The Web Policy Agents are capable of protecting resources that can be hosted on the web servers that they are installed on. This includes any resource that can be represented as a URI that is available on the protected web server. Such a protected URI can be resolved by the web server to static content files such as HTML files or

dynamic content generation programs such as CGI scripts or servlets hosted by an embedded servlet engine. In other words, before a request is evaluated by the web server, the Web Policy Agent can evaluate the necessary credentials of a user and can allow or deny access for the requested resource. Once the request is granted access to the resource, it can be processed internally by the web server as applicable. In other words, the Web Policy Agent uses the request URL to enforce all policy decisions regardless of what that URL maps to internally in the web server. In cases where the request URL maps to a servlet which in turn invokes other servlets or JSPs, the Web Policy Agent will not intercept these subsequent resource requests unless such invocation involves a client side redirect.

On the other hand, a J2EE Policy Agent is capable of protecting web and enterprise applications hosted by the application server on which it is installed. These applications may include resources such as HTML pages, servlets, JSPs, and EJBs. Apart from these resources, any resource that can be accessed as a URI within a protected web application can also be secured by such agents. For example, images that are packaged within a web application can also be protected by the J2EE Policy Agents. These agents allow the evaluation of J2EE policies and can also enforce Identity Server based URL policies like a Web Policy Agent on the resources being requested by the user. Minimally the enforcement is done at the outermost requested URL, but can also be done on any intermediate URLs being chained to this resource on most application servers.

Default Scope of Protection

When installed, a Web Policy Agent automatically protects the entire set of resources available on the web server. However, in order to protect resources within a web application hosted on an application server, the web application must be configured to use the J2EE Policy Agent. Thus if multiple web applications are hosted on an application server on which a J2EE Policy Agent has been installed, only the web applications that have been specifically configured to use the J2EE Policy Agent will be protected by the agent. Other applications will remain unprotected and can potentially malfunction if they depend upon any J2EE security mechanism.

Further, the J2EE Policy Agent can only protect resources that are packaged within a web or enterprise application. Certain application servers provide an embedded web server that may be used to host non-packaged web content such as HTML files and images. Such content cannot be protected by a J2EE Policy Agent unless it is redeployed as a part of a web application.

Modes of Operation

The J2EE Policy Agents provide an explicit control over various modes of operation. Some of these modes such as `SSO_ONLY` and `URL_POLICY` are also achievable in Web Policy Agents, where as other modes of operation such as `J2EE_POLICY` and `ALL` modes do not apply to Web Policy Agents. In these later modes of operation, the J2EE Policy Agent enforces J2EE declarative policies as applicable for the protected application and also provide support for evaluation of programmatic security APIs available within the J2EE specifications.

Different Configuration Properties

The properties that control the operation of J2EE Policy Agents are different from the properties that control the operation of Web Policy Agents. While some properties facilitate the configuration of same or similar features, they may not share the same name, value or format of specification. Apart from the difference in property names, there may be properties that apply only to one type of agents and not to the other.

Installing the Agent

The previous chapter provided an overview of J2EE Policy Agents and listed the servers and platforms supported by these agents. This chapter describes in detail how to install the agents on the supported servers and platforms.

Topics in this chapter include:

- [Pre-Installation Tasks](#)
- [Launching the Installation Program](#)
- [Using the Installation Program](#)
- [Post-Installation Tasks](#)
- [Customizing the Agent Installation](#)

Pre-Installation Tasks

You must perform certain pre-installation tasks before you begin installing the J2EE policy agents. Some of these tasks are to be performed for all the agents and the rest are specific to the agent you are installing. From this list, read the section [Common Tasks](#) first to understand the generic tasks and then move on to the section relevant to your agent. If the agent-specific sections do not include any tasks pertaining to the agent you are installing, you may directly go to the section [Launching the Installation Program](#).

- [Common Tasks](#)
- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic 6.1 SP2](#)
- [Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1](#)

- [Agent for IBM WebSphere 5.0/5.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)
- [Agent for Sun Java System Application Server 8.1](#)

Common Tasks

1. Make sure that the agent you wish to install is supported on the desired platform and application server combination. Also make sure that the agent being installed is certified to work with the Identity Server version for which you intend to use this agent. The complete list of supported application servers on various platforms is available in the [“Supported Servers”](#).
2. Make sure that you have the correct binaries for installing the agent. The agent is distributed as tar-gzip archive for all UNIX platforms. For Windows 2000 platform, these binaries are provided as a Zip archive. Make sure that you are using the correct binaries for the desired platform.
3. Install the required application server if it is not already installed. Refer to the application server documentation for the details on how best to install the application server for your platform of choice.

NOTE The installation of J2EE Policy Agent is not supported on the application server instance on which Sun ONE Identity Server or Sun ONE Portal Server is installed. However, the installation of the J2EE Policy Agent on a different instance of the application server on the same machine is a supported configuration.

Once you have performed the common pre-installation tasks mentioned above, you must perform the tasks relevant to the agent you are about to install. For detailed instructions specific to your agent, select the appropriate section from the following list:

- [Agent for Sun ONE Application Server 7.0](#)

- [Agent for BEA WebLogic 6.1 SP2](#)
- [Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1](#)
- [Agent for IBM WebSphere 5.0/5.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

Before you begin the installation of the agent for Sun ONE Application Server 7.0, you must make sure that:

- The Administration Server for the application server instance that will be protected by the agent is running, and
- The application server instance that will be protected by the agent is shut down.

Agent for BEA WebLogic 6.1 SP2

Before you begin the installation of the agent for BEA WebLogic 6.1 SP2 Server, you must make sure that the BEA WebLogic Server has been shut down.

Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1

Before you install the policy agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1, you must create a server domain as explained below.

Using the configuration or domain wizard appropriate for your server version and operating system, create a new stand-alone server domain. The configuration wizard launch script/program typically resides in the following directory.

Table 2-1 Location of the Configuration Wizard Program for BEA WebLogic Servers

Platform	Location
BEA WebLogic 7.0 SP2 Server	
On Unix platform	<i>bea-home</i> /weblogic700/common/bin/dmwiz.sh
On Windows platform	<i>bea-home</i> \weblogic700\common\bin\dmwiz.cmd
BEA WebLogic 8.1 Server	
On Unix platform	<i>bea-home</i> /weblogic81/common/bin/config.sh
On Windows platform	<i>bea-home</i> \weblogic81\common\bin\config.cmd

Once these steps are done, you must shutdown the BEA WebLogic Server before launching the agent installation program.

Agent for IBM WebSphere 5.0/5.1

Before you begin the installation of agent for IBM WebSphere 5.0/5.1 Application Server, you must ensure that the application server instance that will be protected by this agent is running and the Administration application is deployed on this server.

Agent for PeopleSoft 8.3/8.4/8.8

If you are installing the policy agent for PeopleSoft 8.3/8.4/8.8, you must perform the following tasks for setting up PeopleSoft for SSO before you install the agent.

- [Creating DEFAULT_USER](#)
- [Installing PeopleCode](#)
- [Registering PeopleCode for Authentication](#)

Creating `DEFAULT_USER`

1. Invoke the PeopleSoft web application as explained in the following tables and log on as a privileged user.

Table 2-2 Launching PeopleSoft Application

PeopleSoft Application	URL to launch the application	User	Path to Create User	Path to Register PeopleCode
HRMS 8.3	<p><i>protocol://fqdn:port/servlets/iclientservlet/pia_site_name/?cmd=start</i></p> <p>For example: http://pradeep.eng.example.com:8001/servlets/iclientservlet/peoplesoft8/?cmd=start</p>	PS	Home > PeopleTools > Maintain Security > Use > User Profiles	Home > PeopleTools > Utilities > Use > SignOn PeopleCode
FIN 8.4	<p><i>protocol://fqdn:port/psp/pia_site_name/EMPLOYEE/ERP/h/?tab=DEFAULT</i></p> <p>For example: http://pradeep.eng.example.com:8001/psp/ps/EMPLOYEE/ERP/h/?tab=DEFAULT</p>	VP1	Home > PeopleTools > Security > User Profiles > User Profiles	Home > PeopleTools > Security > Security Objects > SignOn PeopleCode
HRMS 8.8	<p><i>protocol://fqdn:port/psp/pia_site_name/EMPLOYEE/HRMS/h/?tab=DEFAULT</i></p> <p>For example: http://pradeep.eng.example.com:8001/psp/ps/EMPLOYEE/HRMS/h/?tab=DEFAULT</p>	PS	Home > PeopleTools > Security > User Profiles > User Profiles	Home > PeopleTools > Security > Security Objects > SignOn PeopleCode

2. Select Add a New Value and create a user.
3. In the User ID field, enter `DEFAULT_USER`.
4. In the Password field, enter the password. You will need this password while installing the agent.

NOTE This user should not have any privileges/roles.

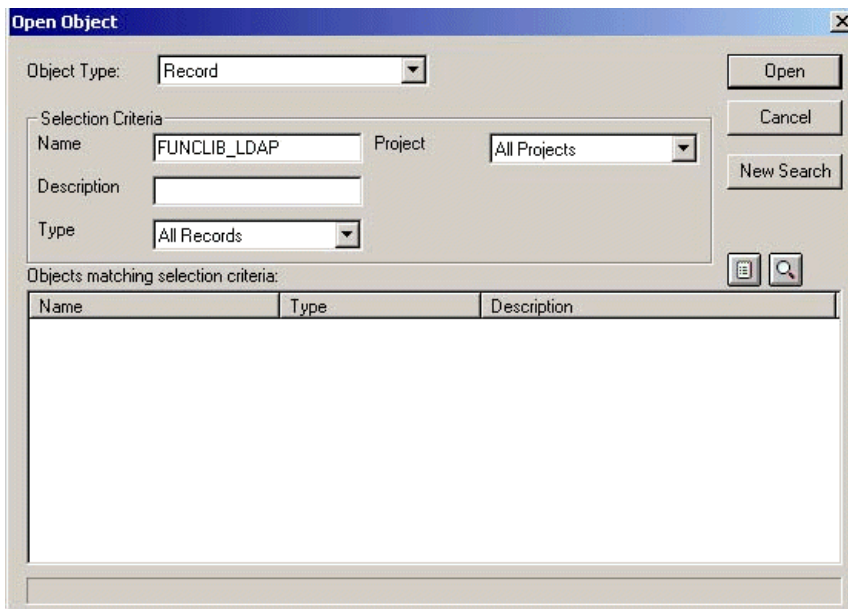
5. Choose the default values or leave the non-mandatory fields blank.

6. In the User Profiles page, open the ID tab and select the ID Type as None. DEFAULT_USER does the initial communication between the PeopleSoft web server servlets and the PeopleSoft Application Server.
7. Click the Save button to save the changes.
8. Log out of the PeopleSoft application.

Installing PeopleCode

1. Invoke Application Designer on the Windows NT machine where PeopleSoft application is installed. To do this, go to Start menu > Programs > PeopleSoft Installation > Application Designer.
2. Log on as a privileged user, who has write permissions to FUNCLIB_LDAP record. For example, PS(8.3) / VP1(8.4) / PSHC(8.8).
3. Navigate to the FUNCLIB_LDAP record. To do this:
 - o Click File > Open.

Figure 2-1 Open Object window



- o In the Open Object window, select the Object Type Record.
- o In the name field, enter FUNCLIB_LDAP.

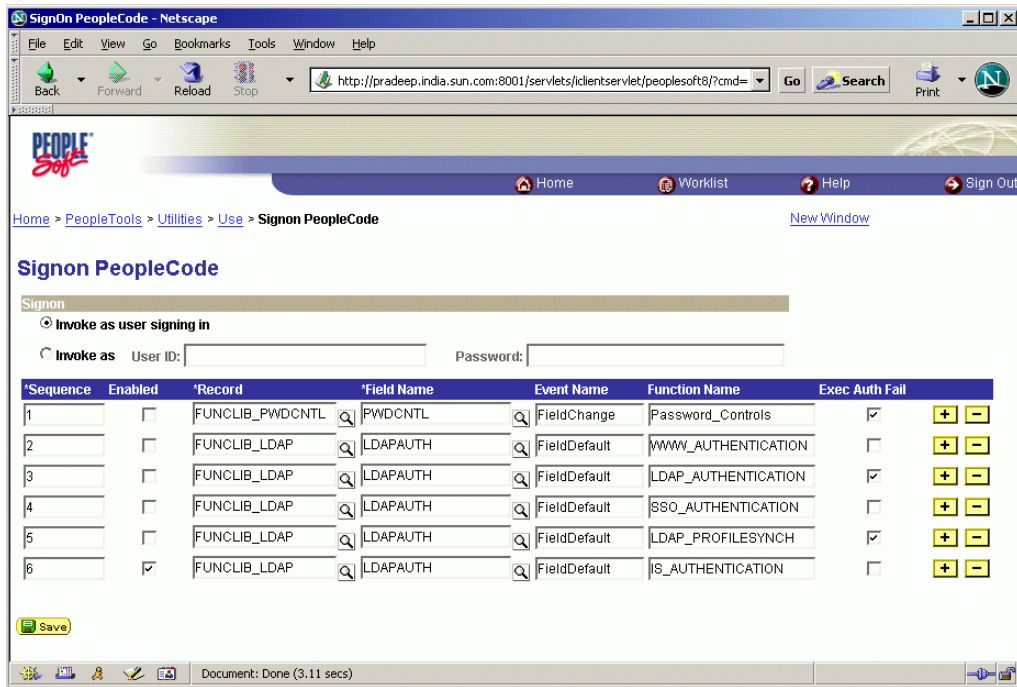
- Leave the default values in the rest of the fields and click Open.
- 4. Select the LDAP Auth row, right-click the mouse button and select View PeopleCode option.
- 5. Append the contents of `AmPeopleCode.txt` to the end of the existing PeopleCode source. This file is provided with the agent binaries.
- 6. Search for Function `getWWWAuthConfig()` in the PeopleCode:


```
Function getWWWAuthConfig()
    &defaultUserId = "";
End-Function
```
- 7. Modify the variable `&defaultUserId` to return the User ID created in previous step as follows:


```
Function getWWWAuthConfig()
    &defaultUserId = "DEFAULT_USER";
End-Function;
```
- 8. Click the Save button to save and exit from the Application Designer.

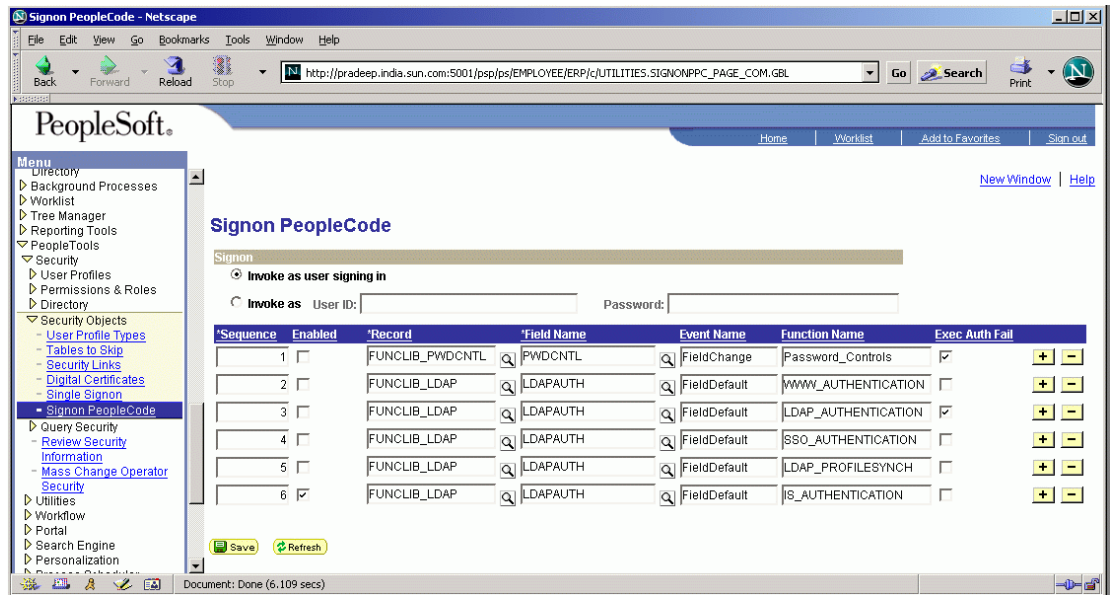
Registering PeopleCode for Authentication

1. Invoke the PeopleSoft web application and log on as a privileged user.
2. Navigate to the SignOn PeopleCode window as explained here:
If you are using PeopleSoft 8.3, go to Home > PeopleTools > Utilities > Use > SignOn PeopleCode.

Figure 2-2 SignOn PeopleCode window in PeopleSoft 8.3

If you are using PeopleSoft 8.4/8.8, go to Home > PeopleTools > Security > Security Objects > SignOn PeopleCode

Figure 2-3 SignOn PeopleCode window in PeopleSoft 8.4/8.8



3. Add a new row, which should look like the 6th row shown in Figure 2-2 and Figure 2-3. To add a new row, click + on the last row.
4. Ensure that the Function Name column shows IS_AUTHENTICATION, the Exec Auth Fail column is not selected, and the Enabled column is selected.
5. Click the Save button to save the changes.
6. Log out of the PeopleSoft application.

NOTE If BEA WebLogic Server and PeopleSoft Application Server are installed on two separate machines, you must install the PeopleSoft agent on both the machines.

Agent for Apache Tomcat Server 4.1.27

If you are installing the J2EE policy agent for Tomcat Server, you must perform the following tasks before you install the agent.

- Make sure the environment variable CATALINA_HOME points to the Tomcat Server directory that contains the Tomcat distribution.

- Make sure the environment variable `CATALINA_BASE` is not set, or if set points to the same Tomcat Server directory as `CATALINA_HOME`.

NOTE If the agent is to be configured for multiple server instances, then `CATALINA_BASE` should point to the new Tomcat instance, while `CATALINA_HOME` should point to the original installation that contains the Tomcat Server binaries and libraries.

- Make sure the Tomcat Server is not running before you install the agent. If the server is running, you must stop it using the Tomcat server shutdown script. You can find these scripts at the following locations:

Table 2-3 Location of the Tomcat Server shutdown script

Platform	Location
On Unix	<code># \$CATALINA_HOME/bin/shutdown.sh</code>
On Windows	<code>C:\> %CATALINA_HOME%\bin\shutdown.bat</code>

Agent for Oracle 9iAS R2 and Oracle 10g

Before you begin installing the policy agent for Oracle 9iAS R2 or Oracle 10g, you must stop all the processes related to Oracle 9iAS or Oracle 10g.

NOTE Not stopping all processes related to Oracle such as `emctl`, `opmnctl`, and `dcmtl` can lead to the malfunctioning of the agent resulting in the application or portions thereof becoming inaccessible.

Before installing the agent for Oracle 10g, you must also make sure that the Apache web server is configured to have a fully qualified host name. To ensure this, the Oracle Universal installation program should be started with the following command-line parameter:

```
./runInstaller OUI_HOSTNAME=fully_qualified_host_name
```

If this is not done, the front-end Apache web server will be misconfigured to have a wrong host name in `httpd.conf`. This is a known issue in Oracle 10g Application Server. To get around this problem, you must change the variable `ServerName` in the file `$ORACLE_HOME/Apache/Apache/conf/httpd.conf` to show the fully qualified host name if you had installed the Oracle 10g Application Server without specifying `OUI_HOSTNAME`.

NOTE An incorrect value to the `ServerName` variable in `httpd.conf` file will lead to the malfunctioning of the agent resulting in the application or portions thereof becoming inaccessible.

To learn more about stopping the processes, refer to the Application Servers' Administration Guides at the following URLs:

- Oracle 9iAS R2

http://download-west.oracle.com/docs/cd/A97329_03/core.902/a92171/toc.htm

- Oracle Application Server 10g

http://download-west.oracle.com/docs/cd/B10464_03/core.904/b10376/toc.htm

Agent for SAP Enterprise Portal 6.0 SP2

If you wish to install the agent for SAP Enterprise Portal 6.0 SP2, you must perform the following tasks before starting the agent installation program:

- [Disabling SAP* User](#)
- [Planning User-Mapping Schemes](#)
- [Shutting Down the Enterprise Portal](#)

Disabling SAP* User

When SAP Enterprise Portal 6.0 SP2 is installed, by default the super-administrator user, SAP*, is active in the system. While this user is active, no other users are allowed to log on to the system for security purposes. In order to make the Enterprise Portal installation usable by regular users, this user must be disabled. Follow the steps outlined in the SAP Enterprise Portal 6.0 SP2 administration guide to disable this user before installing the agent.

Planning User-Mapping Schemes

The agent for SAP Enterprise Portal integrates the user base as it exists in the SAP back end with the user base available in Sun ONE Identity Server. This integration is achieved through a mapping scheme that provides three different ways to map the users in these two separate user bases. These schemes are:

Mapping Based on User IDs When installed, the agent defaults to this mode of mapping the Identity Server users to the SAP user base. In this mode, the agent assumes that the user ID for logging a given user onto the Enterprise Portal is the same as that used for authenticating with Sun ONE Identity Server's authentication service.

Mapping Based on an LDAP Attribute In this mode, the agent maps the Sun ONE Identity Server user to a user-id as specified in a named attribute in the profile of the user in Sun ONE Identity Server. The name of the attribute is configurable and can be changed to the applicable value as necessary. When a user accesses the Enterprise Portal, the agent redirects the user to first authenticate against Sun ONE Identity Server's authentication service. When the user is authenticated successfully, the agent reads the value stored in the named attribute for this particular user and uses that value to sign on to the Enterprise Portal.

Mapping based on HTTP Header In this mode, the agent maps the Sun ONE Identity Server user to a user-id as specified in a named HTTP header available in the user's request. When operating in this mode, the agent will treat requests that do not have the specified named header as invalid requests and would display an error message. If however, a value is available for the named header, then the agent uses this value to log the user onto Enterprise Portal and hosted applications.

Shutting Down the Enterprise Portal

Before you begin the installation of the agent for SAP Enterprise Portal, you must first shut down the Enterprise Portal completely. If the Enterprise Portal is not fully shut down before the agent installation, it may cause corruption of key system files and render a part of or the entire system unusable.

Agent for Macromedia JRun 4

If you are installing the J2EE policy agent for Macromedia JRun 4, make sure the Macromedia JRun 4 instance is not running during the installation of the agent. If the server is running, you must stop it using the JRun server script as follows:

```
# jrun_install_directory/bin/jrun stop jrun_instance_name
```

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

The Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1 can be installed for three different system configurations, as follows:

- On a system where only SAP Web Application Server 6.20 SP1 is installed and you intend to protect the hosted applications.
- On a system where SAP Enterprise Portal 6.20 SP2 is installed and you intend to protect the applications deployed on the underlying SAP Web Application Server as well as the Enterprise Portal.
- On a system where SAP Enterprise Portal 6.20 SP2 is installed and you intend to protect only the Enterprise Portal and not any applications that might be deployed on the underlying SAP Web Application Server.

The agent installation program does not distinguish between any of the above scenarios, but detects the configuration of the system and installs the necessary components based on the detection results. Note that if you intend to use only the third scenario—where the agent is being installed solely to protect the Enterprise Portal installation—then this agent is exactly equivalent to the agent for SAP Enterprise Portal 6.20 SP2.

- Pre-installation tasks common to all scenarios:

[Planning User-Mapping Schemes](#)

[Shutting down the SAP Server](#)

- Pre-installation tasks applicable to scenarios where Enterprise Portal is protected by the agent:

[Disabling SAP* User](#)

- Pre-installation tasks applicable to scenarios where applications hosted on SAP Web Application Server need to be protected:

[Readying the Application for Agent Protection](#)

Planning User-Mapping Schemes

The agent for SAP Enterprise Portal and Web Application Server integrates the user base as it exists in the SAP back end with the user base available in Sun ONE Identity Server. This integration is achieved through a mapping scheme that provides three different ways to map the users in these two separate user bases. These schemes are:

Mapping Based on User IDs When installed, the agent defaults to this mode of mapping the Identity Server users to the SAP user base. In this mode, the agent assumes that the user ID for logging a given user onto the Enterprise Portal or applications hosted on Web Application Server is the same as that used for authenticating with Sun ONE Identity Server's authentication service.

Mapping Based on an LDAP Attribute In this mode, the agent maps the Sun ONE Identity Server user to a user ID as specified in a named attribute in the profile of the user in Sun ONE Identity Server. The name of the attribute is configurable and can be changed to the applicable value as necessary. When a user accesses the Enterprise Portal or the hosted applications, the agent redirects the user to first authenticate against Sun ONE Identity Server's authentication service. When the user is authenticated successfully, the agent reads the value stored in the named attribute for this particular user and uses that value to sign on to the Enterprise Portal or hosted applications.

Mapping Based on HTTP Header In this mode, the agent maps the Sun ONE Identity Server user to a user ID as specified in a named HTTP header available in the user's request. If a value is available for the named header, then the agent uses this value to log the user onto Enterprise Portal or hosted applications.

Shutting down the SAP Server

Before you begin the installation of the agent for SAP Enterprise Portal and Web Application Server, you must first shut down the Web Application Server completely. If the Server is not fully shut down before the agent installation, it may cause corruption of key system files and render a part of or the entire system unusable.

Disabling SAP* User

When SAP Enterprise Portal 6.0 SP2 is installed, by default the super-administrator user, SAP*, is active in the system. While this user is active, no other users are allowed to log on to the system for security purposes. In order to make the Enterprise Portal installation usable by regular users, this user must be disabled. Follow the steps outlined in the SAP Enterprise Portal 6.0 SP2 administration guide to disable this user before installing the agent.

Readying the Application for Agent Protection

Before installing the agent, any applications hosted on SAP Web Application Server that must be protected by this agent installation must be removed from the server completely. After the agent installation is complete, follow the post-installation tasks to configure these applications to be protected by the installed agent.

Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal

Before you install the policy agent for WebLogic 8.1 SP2/SP3 Server/Portal, ensure that the WebLogic 8.1 domain associated with the Server/Portal has been shut down. For more on how to create a portal application; please refer to WebLogic Server documentation available at the BEA Systems, Inc. web site:

<http://www.bea.com>.

Agent for Sun Java System Application Server 8.1

Before you install the policy agent for Sun Java System Application Server 8.1, ensure that the following conditions apply.

- The following server is running:
 - Domain Administration Server (DAS) for the application server instance that will be protected by the agent
- Preferably, the following server has been shut down (this is a standard precautionary measure):
 - Application Server instance that will be protected by the agent

Domain Administration Server (DAS) is an integral concept of Sun Java System Application Server 8.1. The following is a brief description of administrative domains and how they pertain to DAS.

Administrative domains provide a basic security structure whereby different administrators can administer specific groups (domains) of application server instances. By grouping the server instances into separate domains, different organizations and administrators can share a single Application Server installation.

Each domain has its own configuration, log files, and application deployment areas that are independent of other domains. If the configuration is changed for one domain, the configurations of other domains are not affected. Each domain has its own DAS, with a unique port number.

For more information on DAS and administrative domains, see the *Sun Java System Application Server 8.1 Administration Guide*.

Launching the Installation Program

After completing the pre-installation tasks, you are now ready to launch the installation program that will install the policy agent on your system. This installation program is platform-specific and should be used as described in this section. Once the installation program has been launched successfully, you may go to the next section, which provides the necessary details on how to use this installation program.

Launching the Installation Program on Solaris, HP-UX, AIX, and Linux

The binaries for the policy agents for Solaris, HP-UX, AIX and Red Hat Advanced Server platforms are provided as tar-gzip archives. Copy the respective archive to the machine where the application server is installed and then perform the following steps to launch the installation program:

1. Login as the `root` user.
2. Unzip the binary archive using the following command:

```
# gzip -dc binaryname.tar.gz | tar xvf -
```

Please ensure that *binaryname* is substituted with the name of the appropriate agent binary.

3. Set your `JAVA_HOME` environment variable to a JDK version 1.3.1 or higher. If your system does not have the required version of JDK, you may either download and install a compatible version of this software from the Java web site <http://java.sun.com/> or use the JDK provided along with the application server.

The installation program provides two types of interfaces—a GUI or a graphical user interface, and an interactive command line interface. In most cases, the GUI installation program can be used for installing the agent. However, in cases such as when you are installing the agent over a telnet session on a remote server and do not have windowing capabilities, then it is recommended that you use the command-line installation program for installing the agent.

If you choose to use the GUI installation program interface, then it is required that you set your `DISPLAY` environment variable to ensure that the GUI installation program window appears on the correct console.

Once you have completed the above steps, you are now ready to execute the `setup` script that launches the agent installation program. This program can be launched in two different modes—a GUI mode which has a rich graphical interface, and an interactive terminal mode that does not require any windowing capability. The following steps may be used to launch the installation program in each of these two modes.

NOTE To launch the installation program on Red Hat Advanced Server 2.1, you must have the Korn shell installed in the system. If the Korn shell is not available, you may launch the installation program directly by using a Java command such as `java -cp . classname`

Launching the Installation Program in the GUI Mode

1. To launch the installation program in the GUI mode, use the following command:

```
# ./setup
```

When launching the program in a GUI mode, it is required that you set your `JAVA_HOME` and `DISPLAY` environment variables correctly as pointed out in the previous steps. If however, you have not set these variables, the `setup` script will prompt you for their values accordingly.

If your `JAVA_HOME` environment variable is not set correctly, the setup script displays the following prompt:

```
Enter JAVA_HOME location (Enter "." to abort):
```

2. At this prompt, you must type the full path to the JDK installation directory that should be used for launching the installation program. Otherwise, enter a period (.) to abort the installation.

If your `DISPLAY` environment variable is not set correctly, the setup script displays the following prompt:

```
Please enter the value of DISPLAY variable (Enter "." to abort):
```

3. At this prompt, you may specify the hostname for the `DISPLAY` variable. Otherwise enter a period (.) to abort the installation.

NOTE

- In case you enter a value for the `DISPLAY` environment variable which is not appropriate, the installation program window may get displayed on a different console, giving the impression that the installation program is hanging. If such a condition occurs, please verify your `DISPLAY` environment variable by launching any other graphical programs such as `xterm` etc.
 - In the case when the supplied value for `DISPLAY` environment variable is not a valid or reachable hostname, or is that of a host which does not allow you to use its windowing capabilities, the Installation program will automatically shift to the interactive command line mode.
-

Launching the Installation Program in the Command-Line Mode

1. To launch the installation program in a non-GUI or command line mode, use the following command:

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

NOTE

Please note that the installation program command and parameters are case-sensitive. So, if you type a parameter in the upper or the mixed case, the installation program will ignore the parameter and open in the GUI mode.

When launching the program in a command line mode, it is required that you set your `JAVA_HOME` environment variable correctly as pointed out in the previous steps. If however, you have not set this variable, the `setup` script will prompt you for its value as follows:

```
Enter JAVA_HOME location (Enter "." to abort):
```

2. At this prompt, you must type the full path to the JDK installation directory that should be used for launching the installation program. Otherwise, enter a period (.) to abort the installation.

Depending upon the mode in which you have launched the installation program, you should see the appropriate interface appear at this stage.

NOTE

Instead of using the provided script to launch the installation program, you may alternatively invoke the installation program class file included in the agent binaries with a JDK runtime version 1.3.1 or above to launch the installation program. However, this is not a recommended approach.

Launching the Installation Program on Windows

The binaries for the policy agents on Windows platform are provided as a zip archive. Copy this archive to the machine where the application server is installed and follow these steps to launch the installation program:

1. Log into your Windows system as a user with Administrative privileges. If you do not have administrative privileges, either log on as Administrator user or request such privileges to be granted to your account by the system administrator of the machine or domain as applicable.
2. Unzip the agent binaries in a convenient location using any standard Zip utility. The binary contains two executable files `setup.bat` and `install.exe`, which may be used to launch the installation program. Each of these files provides different features for launching the installation program. You may choose either of these two files depending upon your installation requirements.

Using the `setup.bat` file to launch the installation program

In order to use the `setup.bat` file to launch the installation program, you must have a JDK version 1.3.1 or higher available in your system path.

1. To verify the JDK version on your machine, type the following command in a command prompt window:

```
C:\> java -version
```

You will see the following display if your machine has JDK version 1.3.1 or higher.

```
java version 1.3.1_02
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_02-b02)
Java HotSpot(TM) Client VM (build 1.3.1_02-b02, mixed mode)
```

2. To execute `setup.bat`, type the file name at the command prompt window in the directory where it is present, or double-click the file in Windows Explorer. For example:

```
C:\>setup.bat
```

The installation program provides two types of interfaces—a GUI and an interactive command line interface. You can launch the installation program in the GUI mode by invoking `setup.bat` file from a command prompt window as shown above or by double clicking it in Windows Explorer. The installation program may be launched in a command line mode by passing the argument `-nodisplay` to the `setup.bat` script as follows:

```
C:\>setup.bat -nodisplay
```

Using the `install.exe` file to launch the installation program

Using `install.exe` relieves you from the task of setting up your environment path to include a valid version of JDK. This program first checks your system for a compatible JDK version and uses the one that is found. However, if no compatible version is found, this program installs the necessary runtime environment and uses it to launch the installation program.

You can invoke `install.exe` either from the command prompt or by double clicking the file from Windows Explorer. One limitation with using `install.exe` is the fact that it does not recognize any command line arguments the way the `setup.bat` does. Consequently, it is not possible to use `install.exe` to launch the installation program in any mode other than the default GUI mode.

NOTE

- Since the command-line installation program cannot be launched using `install.exe` in cases where the command line installation program is required, it is recommended that you use `setup.bat` to launch the command line installation program.
 - Instead of using the provided scripts and executable that launch the installation program, you may alternatively invoke the installation program class file included in the agent binaries with a JDK runtime version 1.3.1 or above to launch the installation program yourself.
-

Depending upon the mode in which you have launched the installation program, you should see the appropriate interface appear at this stage.

Using the Installation Program

As mentioned in the last section, the agent installation program provides two types of interfaces - a GUI or graphical user interface, and a non-GUI or console-based interactive interface. Once the installation program is launched, you must provide all the necessary information requested by this program in order to successfully install the agent on your system. The following two sections describe in detail how to use this installation program in each interface to successfully install the agent on your system.

Once the installation program has been launched, the expected interaction will be the same on any platform for a given application server agent. Thus the steps outlined here for a certain application server agent installation will be applicable to the same agent installation for the same application server on a different platform.

NOTE

- The information regarding the use of the agent installation program as outlined in this section is applicable to all application server agents supported in this release. Any agent specific differences have been identified here separately.
 - It is important that you perform all the pre-installation tasks relevant to the agent you are installing. These tasks are presented in the section [Pre-Installation Tasks](#).
-

Using the GUI Installation Program

When the agent installation program is launched in the GUI mode, it presents the user with a series of screens that gather the necessary information and report the status of the installation progress and result at certain stages of the overall installation. These screens also contain some navigation buttons that help the user to go forward or backward as necessary and also allow the user to access help messages if needed. The agent installation program also provides active feedback, by means of pop-up windows that contain the necessary help or error messages, to help the user enter correct information in case the provided information is not valid.

Follow these steps to install the agent in the GUI mode:

1. Launch the installation program in the GUI mode as explained in the earlier sections. The installation program begins with a welcome screen. Read the information provided on this screen.

2. Click Next to open the License Agreement text. You must read and understand all the terms and conditions as detailed in this agreement. If you do not agree with any term or condition of this agreement, click the **NO** button to terminate the installation of the agent. If you have read and agree to the terms and conditions, click the **Yes (Accept License)** button to continue.
3. In the Select Installation Directory screen, select the directory in which the agent will be installed. You must specify the full path to the directory of your choice. If you select a directory that does not exist on your system, the installation program will prompt you to specify if the new directory should be created. You can either choose to create this new directory by clicking the **Create Directory** button or select a new directory by clicking the **Choose another Directory** button.

Figure 2-4 Select Installation Directory screen



4. In the Identity Server Details screen, enter the relevant information as requested by the installation program.

Figure 2-5 Identity Server Details screen

Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall Wizard

Identity Server Details Panel

Enter the server information where the Sun(tm) ONE Identity Server is installed.

Sun ONE Identity Server Services Host: buyer.eng.example.com

Sun ONE Identity Server Services Port: 58080

Sun ONE Identity Server Services Protocol: http https

Sun ONE Identity Server Services Deployment URI: /amserver

Sun ONE Identity Server Console Host: buyer.eng.example.com

Sun ONE Identity Server Console Port: 58080

Sun ONE Identity Server Console Protocol: http https

Sun ONE Identity Server Console Deployment URI: /amconsole

amAdmin Password: ****

Re-enter Password: ****

amldapuser Password: ****

Re-enter amldapuser Password: ****

Back Next Exit Help

Sun ONE Identity Services Host: Enter the fully qualified host name of the system on which the Sun ONE Identity Server is installed.

Sun ONE Identity Server Services Port: Enter the port number for the web container used by Sun ONE Identity Server to provide its services.

Sun ONE Identity Server Services Protocol: Select the appropriate protocol that will be used by the agent when communicating with Sun ONE Identity Server services. This protocol value may either be HTTP or HTTPS.

Sun ONE Identity Server Services Deployment URI: Enter the URI that will be used by the agent to access various Sun ONE Identity Server services. The default URI is /amserver.

Sun ONE Identity Server Console Host: Enter the fully qualified hostname of the system on which the Sun ONE Identity Server console is installed.

Sun ONE Identity Server Console Port: Enter the port number of the web container that is used by Sun ONE Identity Server console.

Sun ONE Identity Server Console Protocol: Select the appropriate protocol that will be used by the agent when communicating with Sun ONE Identity Server console. This protocol value may either be `HTTP` or `HTTPS`.

Sun ONE Identity Server Console Deployment URI: Enter the URI that will be used by the agent to access the Sun ONE Identity Server console. The default URI is `/amconsole`.

amAdmin Password: Enter the password assigned to the `amAdmin` user as specified during Sun ONE Identity Server installation. The password entered here should be the `amAdmin` user password originally used during the installation of Sun ONE Identity Server. Even if after the installation of Sun ONE Identity Server, the password for the `amAdmin` user has been changed, you should still enter the original password and not the changed password.

Re-enter Password: Re-enter the `amAdmin` password in this field to ensure that the correct value is used by the agent.

amldapuser Password: Enter the password assigned to the Sun ONE Identity Server internal LDAP authentication user (`amldapuser`) as specified during Sun ONE Identity Server installation.

Re-enter amldapuser Password: Re-enter the `amldapuser` password in this field to ensure that the correct value is used by the agent.

NOTE

- In situations where you are not using the features of the agent that require LDAP connectivity, you may choose to leave the `amAdmin` password blank. By doing so, the value for this password will be set to `changeit`, which can be changed at a later stage as necessary.
- If you select the `HTTPS` protocol as either Sun ONE Identity Server Services Protocol, or Sun ONE Identity Server Console Protocol, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for the application server's JDK keystore. Please refer to the application server documentation to learn how this can be done.

-
5. In the Directory Server Details screen, enter the necessary information regarding the Directory Server used with Sun ONE Identity Server.

Figure 2-6 Directory Server Details screen

Directory Server Details Panel

Enter the directory information corresponding to the Sun(tm) ONE Identity Server.

Directory Host: buyer.eng.example.com

Directory Port: 389

SSL Enabled:

Root Suffix: dc=Example,dc=COM

Installation Organization: dc=Example,dc=COM

Back Next Exit Help

Directory Host: Enter the fully qualified hostname of the system where the Directory Server is installed.

Directory Port: Enter the port number used by this Directory Server.

SSL Enabled: Select this check box if the Directory Server uses SSL to communicate on the said port.

Root Suffix: Enter the root suffix to be used for accessing information stored in this Directory Server.

Installation Organization: Enter the complete DN of the default organization that was created when Sun ONE Identity Server was installed.

NOTE If you select the SSL Enabled checkbox, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for the application server's JDK keystore. Please refer to the application server documentation to learn how this can be done.

6. In the Agent Core Settings Details screen, enter the key information necessary for the agent to function correctly.

Figure 2-7 Agent Core Settings Details screen

Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall Wizard

Sun
microsystems

Sun ONE
Identity
Server

Agent Core Settings Details Panel

Enter Agent configuration values.

Agent Host Name: nila.eng.example.COM

Preferred protocol listening port: 81

Server's preferred protocol: http https

Config Reload Interval: 10

Enable Not-Enforced List Cache:

Number of Entries in Cache: 1000

Cache Expiration Time in Seconds: 60

Primary Application Context Path: /TestSuite

Agent Filter Mode: J2EE_POLICY

Access Denied URL:

Maximum allowed Login Attempts: 5

Back Next Exit Help

Agent Host Name: Enter the fully qualified hostname on which the application server protected by the agent is installed.

Preferred protocol listening port: Enter the preferred port number on which the application server provides its services.

Server's preferred protocol: Select the protocol used by the application server to provide its services on the given port number as entered for the field Preferred protocol listening port.

Config Reload Interval: Enter the amount of time in seconds after which the agent will automatically reload any changes made to the its configuration. Specify the value 0 to disable this feature.

The agent supports hot-swap configuration, which allows changes to take effect without having to restart the application server. This feature can be very helpful in a controlled development and test environment where frequent changes to the configuration are needed to arrive at the correct configuration settings. It is recommended that this feature be disabled for production systems to ensure the optimal use of system resources and to avoid accidental

changes to be picked up by the agent. Also note that although the majority of agent configuration is hot-swap enabled, there are some configuration settings that require a complete application server restart. Please refer to [Chapter 3, “Agent Configuration”](#) on page 153 to learn more about this feature.

Enable Not-Enforced List Cache: Select this check-box if you wish to enable the caching of Not-Enforced List entries as evaluated by the agent against incoming user requests.

It is recommended that the Not-Enforced List caching be enabled in order to optimize system performance. However, the overall system performance can be affected if the corresponding values of Cache Size and Cache Expiration time are not suited for your deployment. It is therefore recommended that the values used for these configuration settings be carefully evaluated in a controlled testing environment before being used in production. It should also be noted that the agent maintains two caches in its memory—one for recording the URIs that were evaluated as enforced and the other for recording the URIs that were evaluated as not-enforced. The specified values of Number of Entries in Cache and Cache Expiration time are equally applicable to both of these caches. This factor must be considered when setting the values for the size and expiration time of the cache.

Number of Entries in Cache: Enter the size of cache used by the agent to store the evaluated results.

Cache Expiration Time in Seconds: Enter the amount of time in seconds after which the agent can purge an entry from its cache to free up the cache memory for newer entries.

Primary Application Context Path: Enter the context path for the primary application being protected by the agent. If this application is deployed at the root context of the application server, enter the value “/”.

When installing the agent for SAP Enterprise Portal 6.0 SP2, enter the context path as `/irj`.

When installing the agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, you should ensure that the primary application context path is set to the context path of the application that will be available throughout the life of the server. If your SAP installation does not include the

Enterprise Portal, you must specify the value corresponding to one of the applications that will be protected by this agent. If, however, your SAP installation does include the Enterprise Portal, you may use the context path `/irj` as well

In a situation where there are multiple applications deployed on the application server, you must choose the context path of the application that is available for as long as the application server is alive. This value is used by the agent to receive notifications from the Sun ONE Identity Server and is also used to provide support for legacy browsers, etc. Providing an incorrect value for this configuration setting may lead to the malfunctioning of the agent and the application server.

Agent Filter Mode: Select the agent filter mode that you would like to run the agent in. For a complete description of the available agent filter modes, refer to the chapter [Chapter 3, “Agent Configuration” on page 153](#).

When installing the agent for SAP Enterprise Portal 6.0 SP2, the agent filter mode must be set to either `SSO_ONLY` or `URL_POLICY`. Other agent filter modes are not supported for this agent.

Access Denied URL: Enter the complete URI for the access denied page to be used by the agent. If this value is not specified, the agent will use HTTP error status code 403 (`FORBIDDEN`) to deny access to resources as needed.

Maximum Allowed Login Attempts: Enter the number of unsuccessful access attempts in succession after which the user will not be allowed to access the requested URL temporarily for security purposes. Specify the value 0 if you want to disable this feature.

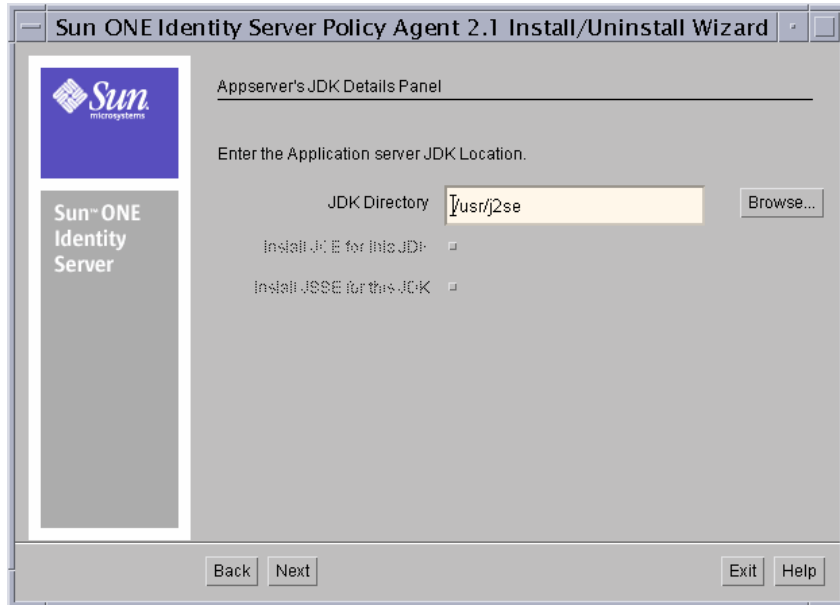
The Login Attempt Limit feature can be used to guard the hosted application from Denial-Of-Service attacks where the end user can overload the application server by repeated authentication requests. By disabling this feature, the system remains vulnerable to such attacks. Therefore this feature should not be disabled unless there is a specific requirement that necessitates the disabling of this feature.

NOTE

If you select `HTTPS` for Server's preferred protocol, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for the Sun ONE Identity Server. Please refer to Sun ONE Identity Server documentation to learn how this can be done.

7. Enter the information regarding the JDK being used by the application server in the Application Server's JDK Details screen. If you are installing the agent for PeopleSoft, you will not get the JDK screen now but at a later stage in the installation process.

Figure 2-8 Application Server's JDK Details screen



JDK Directory: Enter the complete path to the directory where the JDK being used by your application server is installed.

Install JCE for this JDK: Select this box if the JDK being used by your application server does not have JCE installed.

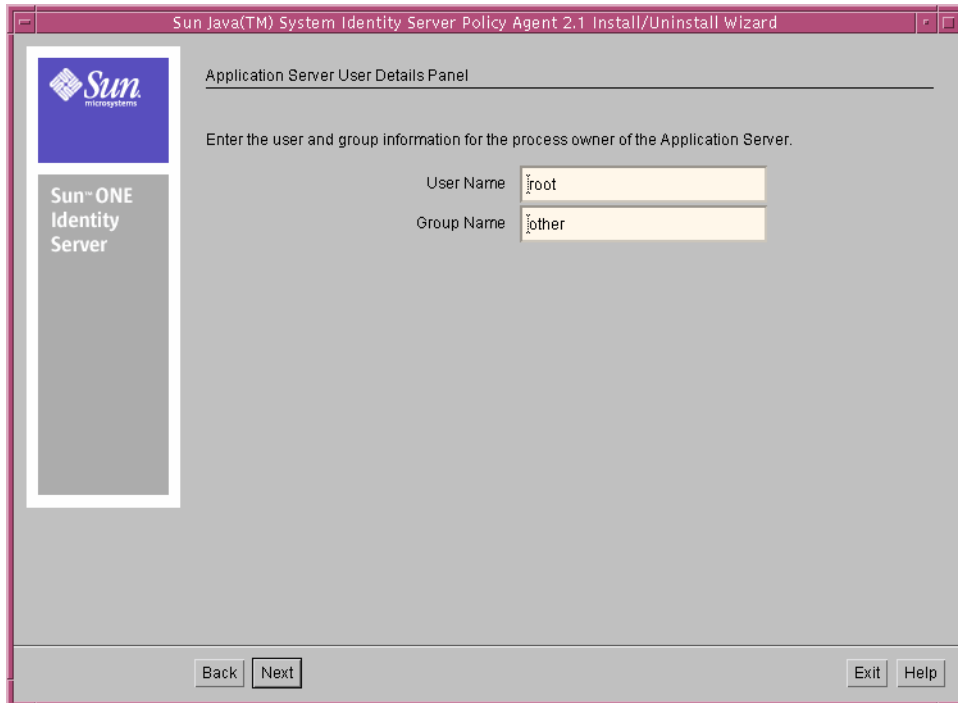
Install JSSE for this JDK: Select this box if the JDK being used by your application server does not have JSSE installed.

NOTE

- If the JDK being used by your application server does not have JCE installed, you must select the Install JCE for this JDK checkbox. Failing to do so can result in the installation program failure.
 - In case your application server is based on JDK 1.4 or above, the option to install JCE and JSSE will be automatically disabled by the agent installation program.
 - If you are installing the agent for BEA WebLogic Server 8.1, you must ensure that both the checkboxes are cleared.
-

8. Enter the user and group information for the process owner of the application server in the Application Server User Details screen. This screen appears only when the agent for Tomcat Server 4.1.27, Oracle 9iAS R2, Oracle 10g, SAP Enterprise Portal SP2 or Macromedia JRun 4 is being installed on a Unix platform.

Figure 2-9 Application Server User Details screen



User Name: The name of the user running the application server process.

Group Name: The name of the group to which the user running the application server process belongs.

NOTE

- Failing to enter the correct user name and group name of the user who owns the application server processes can cause the agent to malfunction.
 - The server instance should be started or stopped only by the process owner of the application server. If these tasks are done by another user, it will cause the agent to malfunction.
-

The screens displayed thus far by the agent installation program are common to all the agents. However, the screens that follow are specific to the application server agent that is being installed.

For detailed instructions for each agent, from the following list, click the appropriate link depending on which application server agent you are installing. After you perform the steps in the respective section, proceed to [“Summary Screen” on page 82](#).

- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic Server 6.1 SP2](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1](#)
- [Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

If you are installing the agent for Sun ONE Application Server 7.0, enter the configuration information on this screen as explained in this section.

Figure 2-10 Sun ONE Application Server Configuration screen

Application Server bin directory: Enter the complete path to the `bin` directory of the Sun ONE Application Server 7.0 installation.

Instance Config Directory: Enter the complete path to the `config` directory of the Sun ONE Application Server 7.0 instance, which will be protected by this agent.

Admin User Name: Enter the user name of the Administrator of this Sun ONE Application Server 7.0 instance.

Admin Server Port: Enter the port number on which the Administration Server for this Sun ONE Application Server 7.0 is available.

Admin Password: Enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

Re-enter Admin Server Password: Re-enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

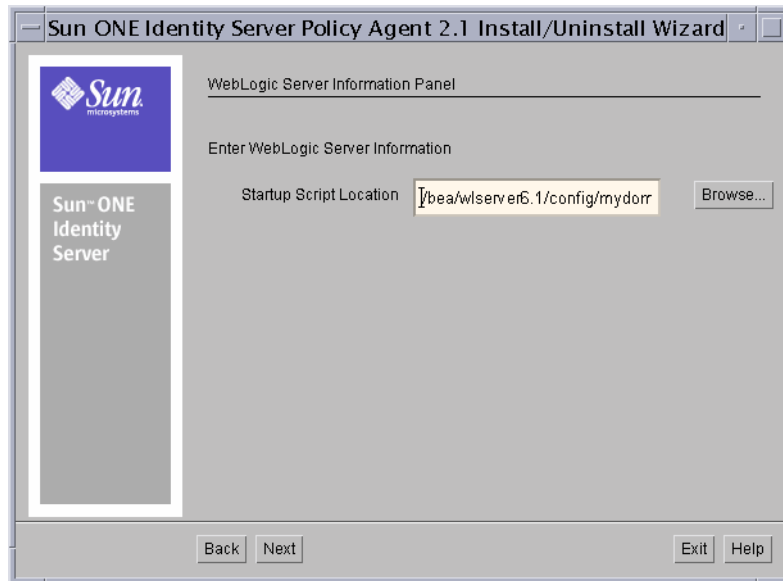
NOTE You must ensure that the Sun ONE Application Server 7.0 Administration Server is available at the time of installation of the agent.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for BEA WebLogic Server 6.1 SP2

If you are installing the agent for BEA WebLogic Server 6.1 SP2, enter the following configuration information on this screen.

Figure 2-11 WebLogic Server Information screen



Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server instance.

- NOTE**
- When the WebLogic Server is in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the CLASSPATH and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of CLASSPATH and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent CLASSPATH to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for IBM WebSphere Application Server 5.0/5.1

If you are installing the IBM WebSphere Application Server instance, enter the following configuration information.

Figure 2-12 WebSphere Application Server Configuration screen

The screenshot shows a configuration window titled "Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall Wizard" with a sub-panel "WebSphere Application Server Config Panel". The panel contains the following fields and controls:

- WebSphere Root Directory:** Text box containing "/opt/WebSphere/AppServer" with a "Browse..." button to its right.
- Instance Config Directory:** Text box containing "/opt/WebSphere/AppServer/cor" with a "Browse..." button to its right.
- WebSphere Admin Connector:** Radio buttons for "SOAP" (selected) and "RMI".
- WebSphere Admin Connector Port:** Text box containing "3880".
- WebSphere Cell Name:** Text box containing "mycomputer".
- WebSphere Node Name:** Text box containing "mycomputer".
- WebSphere Server Instance Name:** Text box containing "server1".

At the bottom of the window are four buttons: "Back", "Next", "Exit", and "Help".

WebSphere Root Directory: Enter the complete path to the root directory for the IBM WebSphere Application Server.

Instance Config Directory: Enter the complete path to the config directory of the IBM WebSphere Application Server instance that will be protected by this agent.

WebSphere Admin Connector: Specify the protocol for connecting to WebSphere Administration Service. You can choose between SOAP and RMI.

WebSphere Admin Connector Port: Enter the Administration Service Port for specified Connector.

WebSphere Cell Name: Enter the cell name for the IBM WebSphere Application Server.

WebSphere Node Name: Enter the node name for the IBM WebSphere Application Server.

WebSphere Server Instance Name: Enter the instance name of the IBM WebSphere Application Server instance.

NOTE

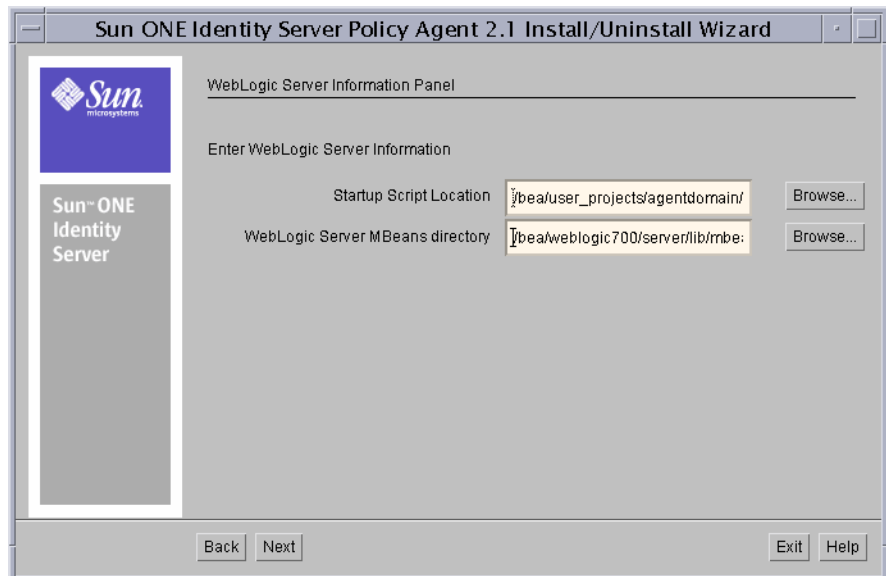
- You must ensure that the IBM WebSphere Application Server 5.0/5.1 instance is running at the time of the agent installation.
 - The agent installation program stops the IBM WebSphere Application Server at the end of the installation. You may manually restart it later.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1

If you are installing the agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1, enter the following configuration information on this screen.

Figure 2-13 WebLogic Server Information screen



Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server instance. This script is typically located under the following directory:

WebLogic_Install_Dir/bea/user_projects/server-domain-name

WebLogic Server MBeans Directory: Enter the full path to the `mbeantypes` directory:

WebLogic_Install_Dir/bea/weblogic/server/lib/mbeantypes

This directory stores MBeans required to deploy on WebLogic.

NOTE

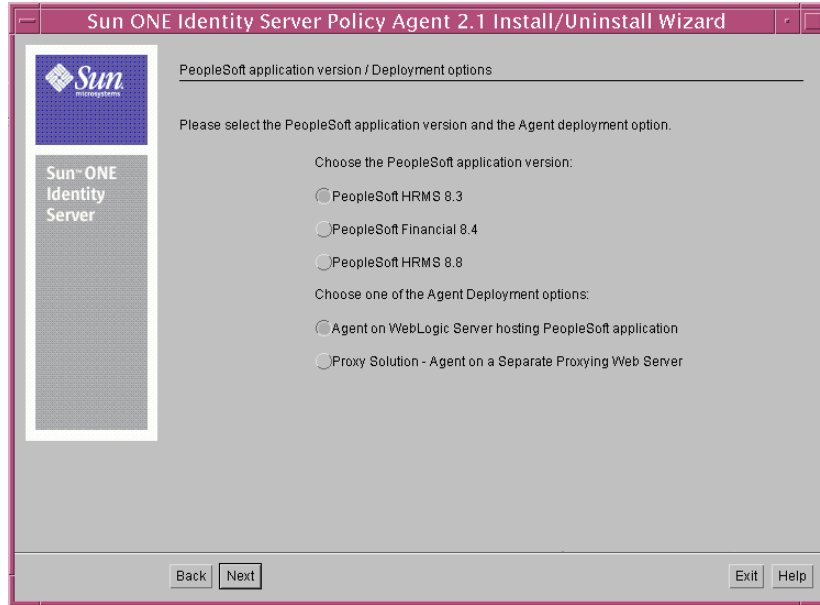
- When WebLogic Server is in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the `CLASSPATH` and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of `CLASSPATH` and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent `CLASSPATH` to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for PeopleSoft 8.3/8.4/8.8

If you are installing the agent for PeopleSoft, perform the following steps:

1. In the PeopleSoft Application Version screen, choose the PeopleSoft application version on which you are going to install the agent.

Figure 2-14 PeopleSoft Application Version/Deployment Options screen

2. Choose one from the following agent deployment options:

Agent on WebLogic Server hosting PeopleSoft application: Select this option to install the agent on top of BEA WebLogic Server hosting the PeopleSoft application.

Proxy Solution - Agent on a Separate Proxying Web Server: Select this option if you want to configure Sun ONE Web Server (also referred as iPlanet Web Server) as a proxy for BEA WebLogic Server hosting PeopleSoft application.

NOTE When installing the agent on PeopleSoft 8.3 HRMS with BEA WebLogic 5.1 front-end server, ensure that the BEA WebLogic 5.1 server has the Service Pack 12 installed. Not having this service pack installed on BEA WebLogic server could result in the malfunction of the PeopleSoft application after the installation of the agent.

To upgrade the BEA WebLogic server to Service Pack 12, you can download the binaries from PeopleSoft distribution site located at <ftp://ftp.peoplesoft.com/outgoing/GSC/jim/weblogic510>. Please follow the instructions provided at this location to upgrade the server.

3. Click Next and in the PeopleSoft Installation Information screen, provide the details about your PeopleSoft installation.

Figure 2-15 PeopleSoft Installation Information screen

Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall Wizard

PeopleSoft Installation information

Enter the PeopleSoft installation information.

PeopleSoft Application Server installed locally

WebLogic Server installed locally

PeopleSoft AppServer Directory Browse...

PeopleSoft Domain Name

WebLogic Instance Directory Browse...

PeopleSoft user ID

PeopleSoft user's primary group

Back Next Exit Help

PeopleSoft Application Server Installed locally: Click this check box if you have PeopleSoft Application Server installed locally.

WebLogic Server Installed locally: Click this check box if you have WebLogic Server installed locally.

NOTE Select one or both of the above options according to your installation. You need to select at least one to proceed.

PeopleSoft AppServer Directory: Enter the full path to the directory where the PeopleSoft Application Server is installed. For example, *PS_HOME/appserv*

PeopleSoft Domain Name: Enter the PeopleSoft domain that this agent will protect. For example, *HDMO*.

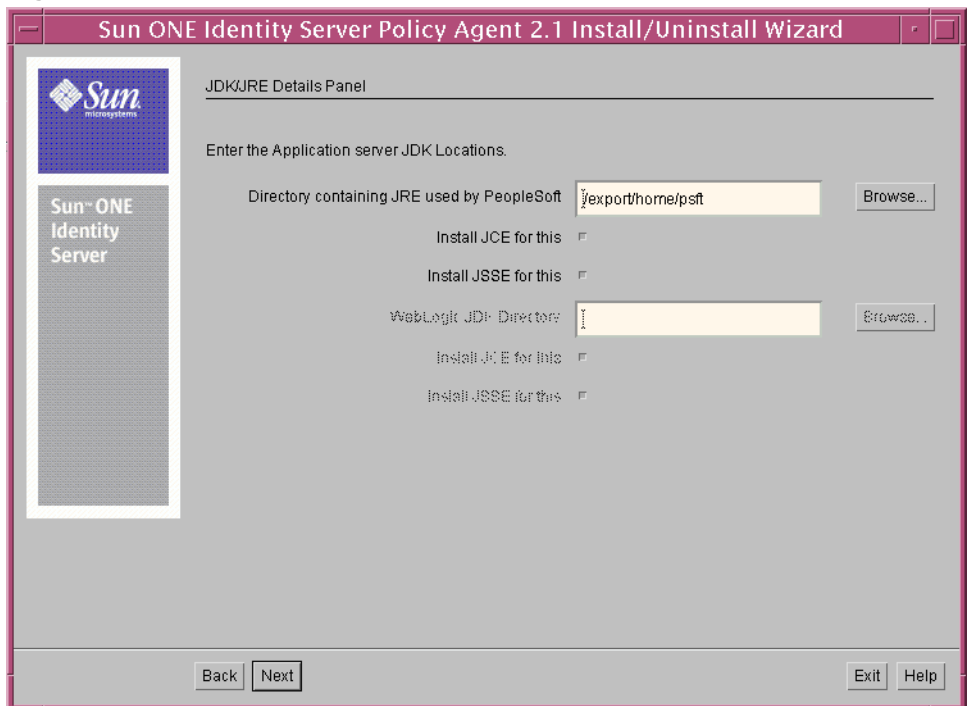
WebLogic Instance Directory: Enter the full path to the WebLogic instance directory. For example, *PS_HOME/weblogic/myserver* or *BEA_HOME/wlserver6.1/config/peoplesoft*

PeopleSoft user ID: Enter the system user ID used to install PeopleSoft software. For example, `psft`

PeopleSoft user's primary group: Enter the primary group of the PeopleSoft user. For example, `sys`

4. Click Next and in the JDK/JRE Details screen, enter the locations of the application server's JDK.

Figure 2-16 JDK/JRE Details screen



Directory containing JRE used by PeopleSoft: Enter the complete path to the directory containing the JRE used by the application server.

Install JCE for this: Choose this option if the Sun JCE provider needs to be installed on the JDK.

Install JSSE for this: Choose this if the Sun JSSE provider needs to be installed on the JDK.

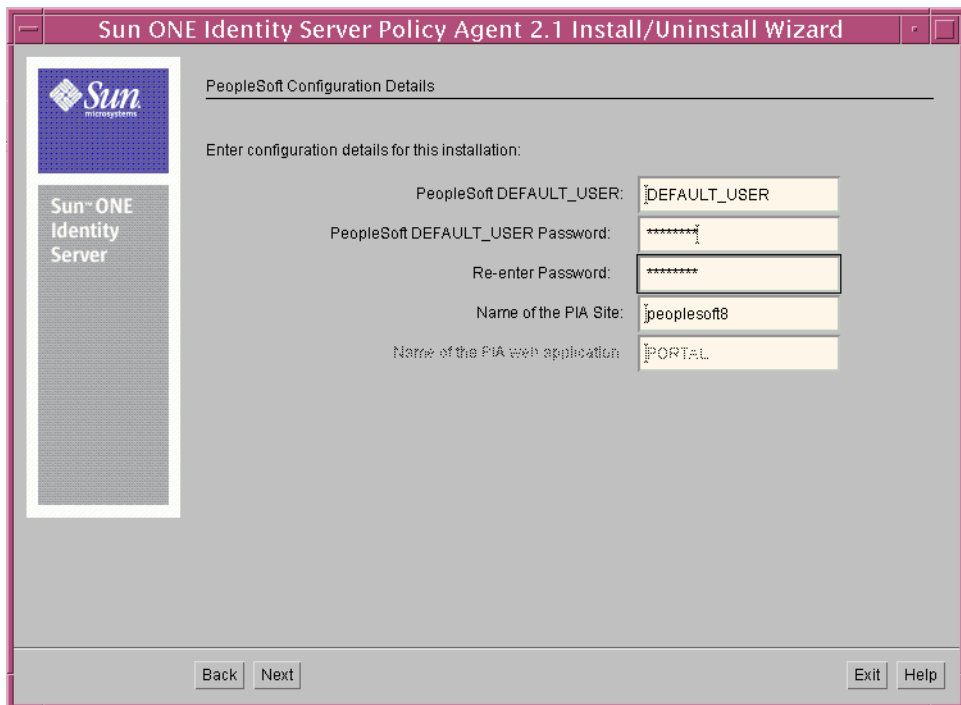
WebLogic JDK Directory: Enter the complete path to the JDK used by WebLogic Server.

Install JCE for this: Choose this option if the Sun JCE provider needs to be installed on the JDK.

Install JSSE for this: Choose this option if the Sun JSSE provider needs to be installed on the JDK.

5. Click Next and in the PeopleSoft Configuration Details screen, enter the configuration details for this installation.

Figure 2-17 PeopleSoft Configuration Details



The screenshot shows a window titled "Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall Wizard". The main content area is titled "PeopleSoft Configuration Details" and contains the instruction "Enter configuration details for this installation:". Below this, there are five input fields with labels: "PeopleSoft DEFAULT_USER:" (containing "DEFAULT_USER"), "PeopleSoft DEFAULT_USER Password:" (containing "*****"), "Re-enter Password:" (containing "*****"), "Name of the PIA Site:" (containing "peoplesoft8"), and "Name of the PIA web application:" (containing "PORTAL"). At the bottom of the window, there are four buttons: "Back", "Next", "Exit", and "Help". On the left side of the window, there is a vertical sidebar with the Sun logo and the text "Sun ONE Identity Server".

PeopleSoft DEFAULT_USER: This field displays the generic PeopleSoft user ID that the web server uses to identify itself to the application server. For example, DEFAULT_USER. For more information on creating this user, see the section "[Creating DEFAULT_USER.](#)"

PeopleSoft DEFAULT_USER Password: Enter the PeopleSoft DEFAULT_USER password assigned while creating the DEFAULT_USER.

Re-enter Password: Enter the password for DEFAULT_USER again for confirmation.

Name of the PIA Site: Enter the name of the PIA site, as given during PeopleSoft installation. For example, peoplesoft8

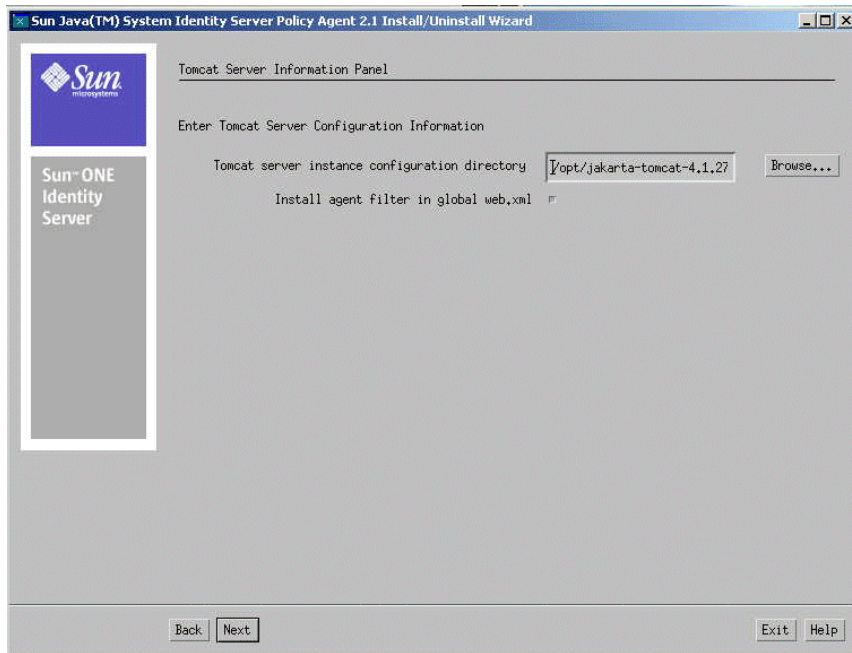
Name of the PIA web application: Enter the name of the PIA web application as given during PeopleSoft installation. For example, PORTAL.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for Apache Tomcat Server 4.1.27

If you are installing the agent for Apache Tomcat Server 4.1.27, enter the configuration information in the Tomcat Server Information screen as explained in this section.

Figure 2-18 Tomcat Server Information panel



Tomcat server instance configuration directory: Enter the complete path to the configuration directory of the Tomcat Server instance.

Install agent filter in global web.xml: Choose this option if you want to install the agent filter in the global deployment descriptor of Tomcat Server. This will enforce Sun ONE Identity Server policies on the entire instance including its index page, if any. Moreover, you will not be required to manually add filter definition for any individual web application deployed on this instance.

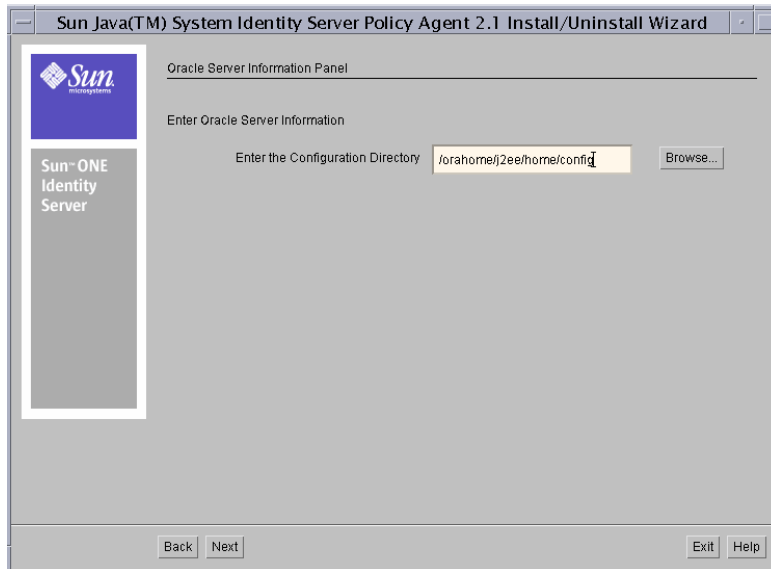
NOTE There is more than one way to deploy the agent filter. To see details about both these options, see the section [Agent for Apache Tomcat Server 4.1.27](#) under [Post-Installation Tasks](#).

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for Oracle 9iAS R2 and Oracle 10g

When you are installing the agent for Oracle 9iAS R2 or Oracle 10g, the installation program displays the Oracle Server Information panel where you must enter the following information.

Figure 2-19 Oracle Server Information screen



Enter the configuration directory: Enter the full path to the configuration directory of the OC4J instance. This directory is typically located as follows:

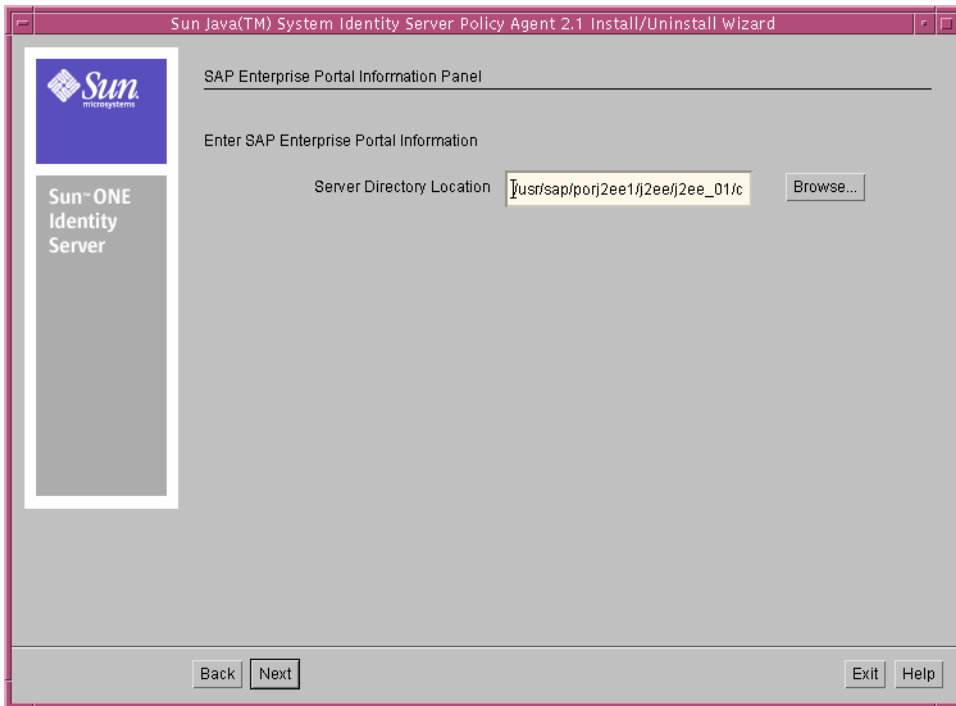
- Default OC4J Instance
ORACLE_HOME/ora9ias/j2ee/home/config
- Sample OC4J Instance
ORACLE_HOME/ora9ias/j2ee/test/config

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for SAP Enterprise Portal 6.0 SP2

If you are installing the agent for SAP Enterprise Portal 6.0 SP2, enter the configuration information on this screen as explained in this section.

Figure 2-20 SAP Enterprise Portal Information screen



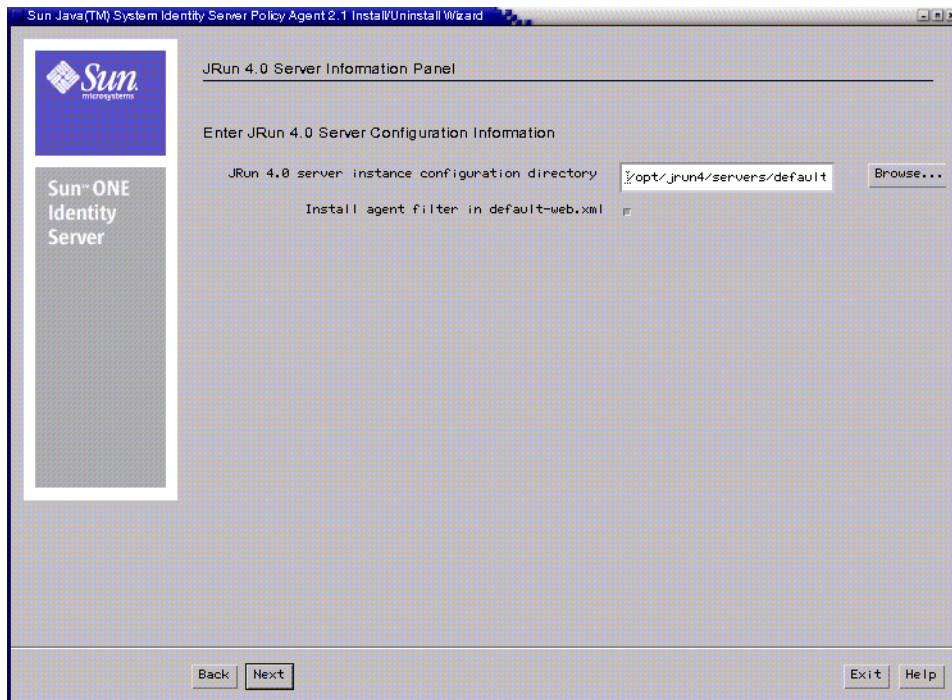
Server Directory Location: Enter the complete path to the installation directory of the server that will be protected by this agent. Use the browse button to locate this directory if necessary.

NOTE You must ensure that the SAP Enterprise Portal server is not running during the agent installation.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for Macromedia JRun 4

If you are installing the agent for Macromedia JRun 4, enter the configuration information in this screen as explained in this section.

Figure 2-21 JRun Server 4.0 Information panel

JRun 4.0 server instance configuration directory: Enter the complete path to the SERVER-INF directory of the Macromedia JRun server instance.

Install agent filter in default-web.xml: Choose this option if you want to install the agent filter in the global deployment descriptor of Macromedia JRun server.

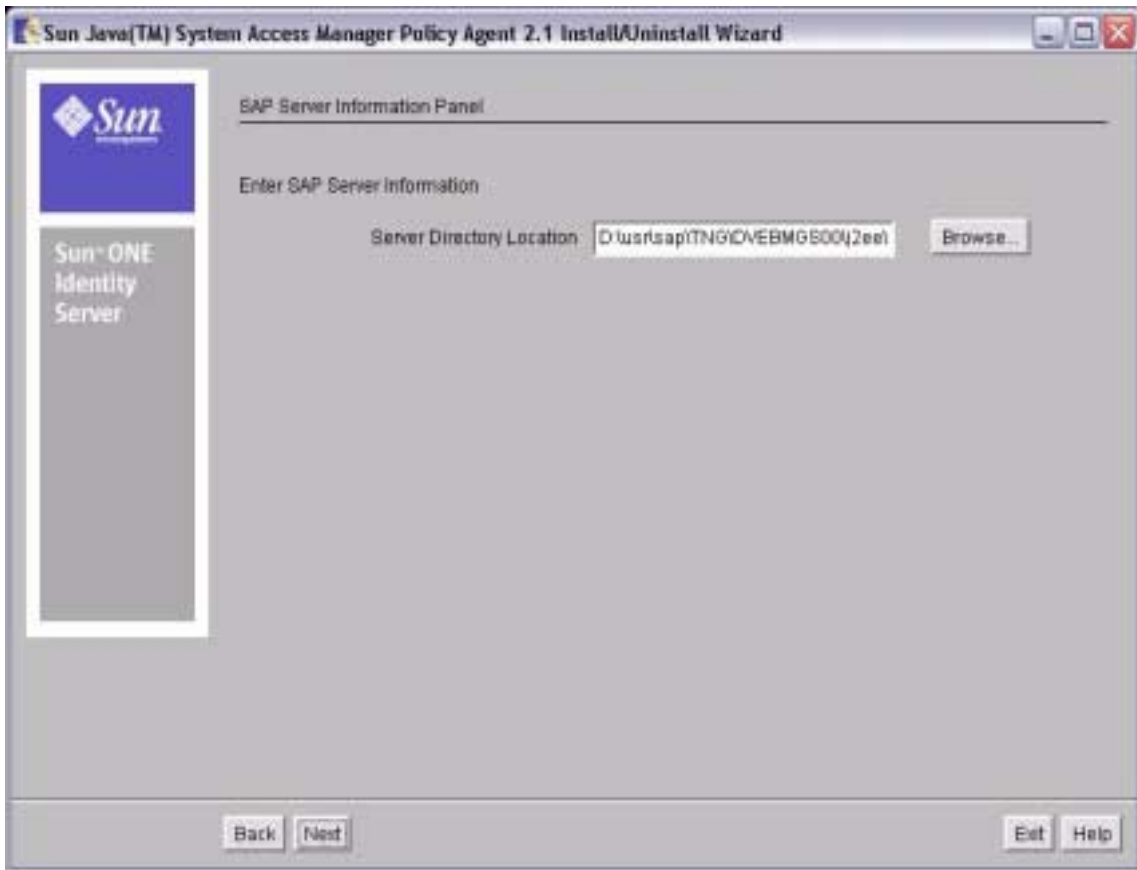
NOTE If you select this option, all resources in the server instance will be protected by the servlet filter defined in `default-web.xml` and users will not be able to access even the welcome page of the server, if any, without providing the credentials and adding a rule to access that resource in Sun ONE Identity Server. One way to handle this problem is to add your server's welcome page URL or any other resource that does not need the protection, to the property `com.sun.am.policy.amFilter.notenforcedList[<index>]` in the agent configuration file `AMAgent.properties`.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

If you are installing Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, enter the configuration information on the SAP Enterprise Portal and Web Application Server information screen as explained in this section.

Figure 2-22 SAP Enterprise Portal and Web Application Server Information Screen



Server Directory Location Enter the complete path to the installation directory of the server that will be protected by this agent. Use the browse button to locate this directory if necessary.

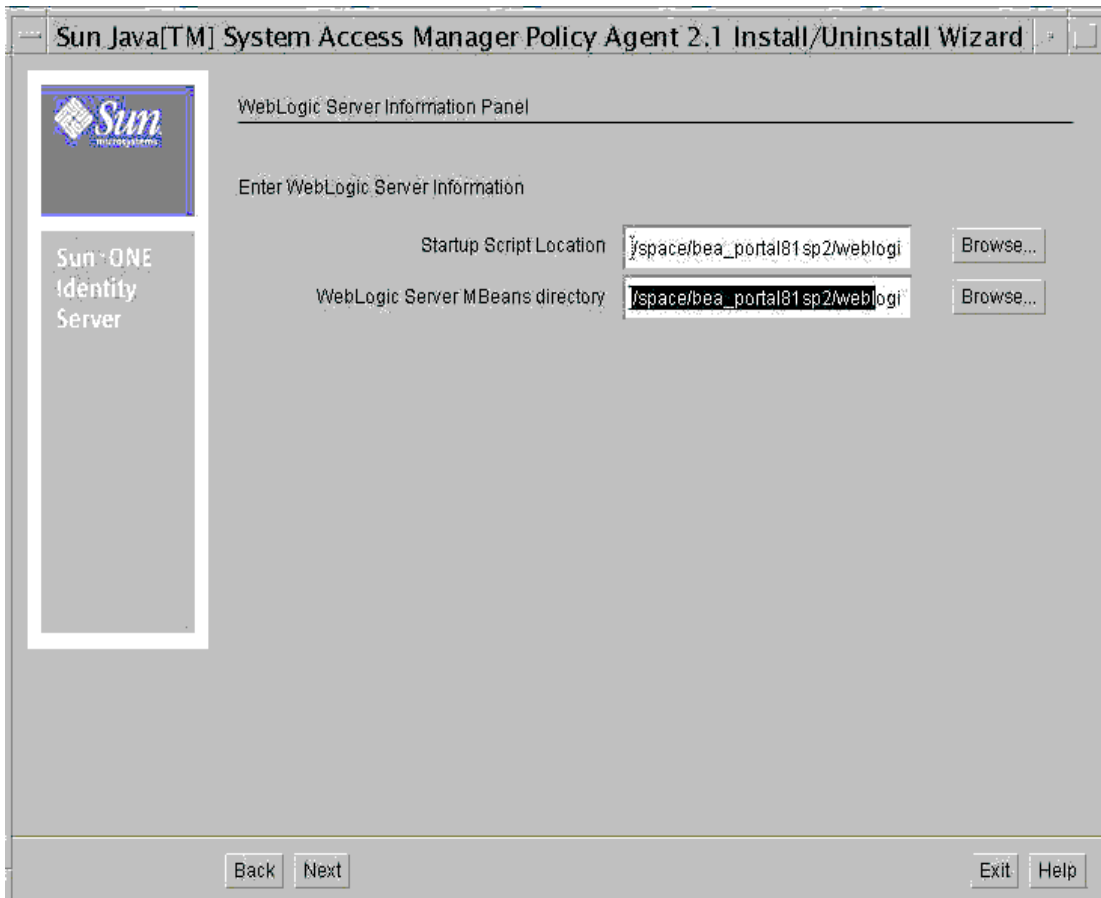
NOTE You must ensure that the SAP server is not running during the agent installation.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Screen](#) section.

Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal

If you are installing the agent for WebLogic 8.1 Server/Portal, enter the configuration information on the WebLogic 8.1 information screen as explained in this section.

Figure 2-23 BEA WebLogic 8.1 Server/Portal Information Screen



Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server/Portal instance. This script is typically located under the following directory:

WebLogic_Portal_Install_Dir/user_projects/portal-domain-name

WebLogic Server MBeans Directory: Enter the full path to the mbeantypes directory:

WebLogic_Portal_Install_Dir/weblogic81/server/lib/mbeantypes

This directory stores MBeans required to deploy on WebLogic.

NOTE

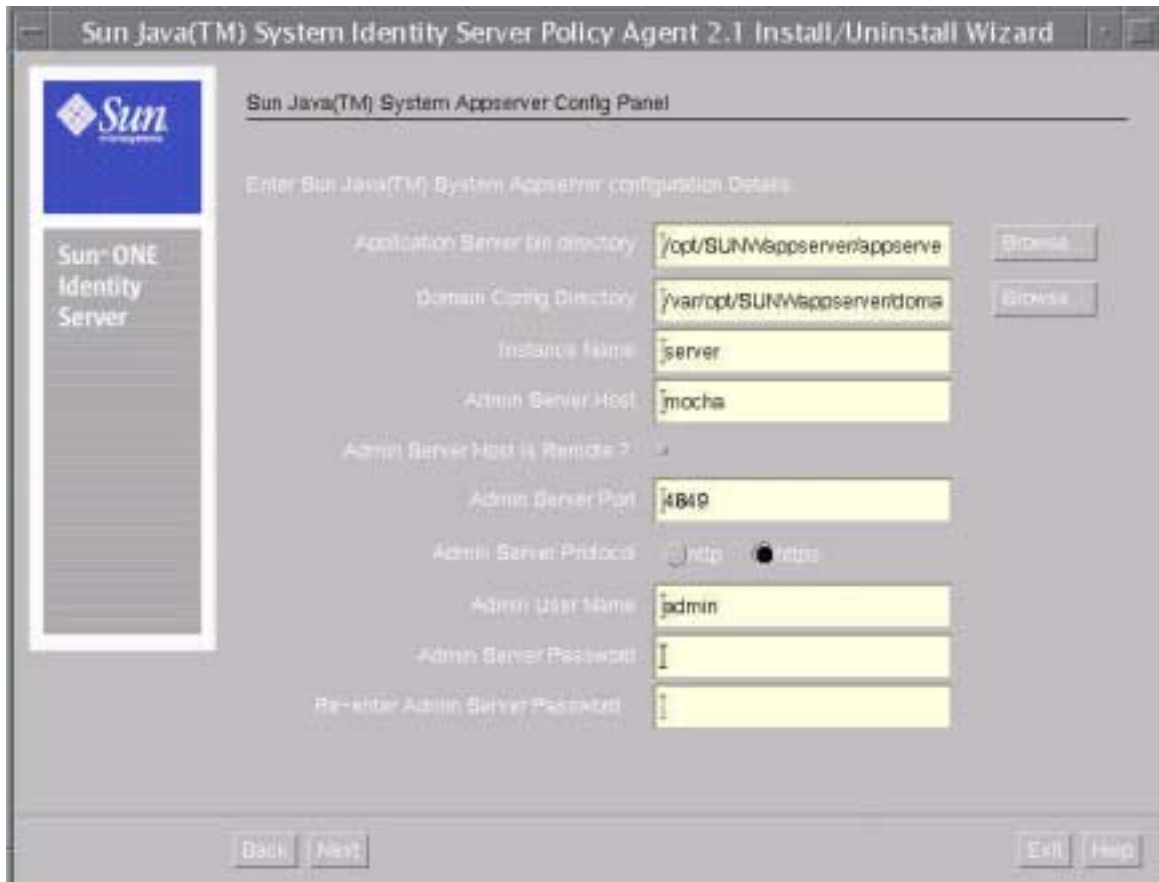
- When the WebLogic Portal is in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the CLASSPATH and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of CLASSPATH and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent CLASSPATH to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed the [Summary Screen](#) section.

Agent for Sun Java System Application Server 8.1

If you are installing the agent for Sun Java System Application Server 8.1 enter the configuration information on the screen as explained in this section.

Figure 2-24 Sun Java System Application Server 8.1 Information Screen



As explained previously, Sun Java System Application Server 8.1 makes use of administrative domains.

Each domain has its own configuration, log files, and application deployment areas that are independent of other domains. Therefore, each domain has its own Application Server instance, which is referred to as Domain Administration Server (DAS).

During the installation you are prompted for the server instance that the agent will protect and the corresponding DAS information.

Application Server Bin Directory: Enter the complete path to the `bin` directory of the Sun Java System Application Server 8.1 installation.

Domain Config Directory: Enter the complete path to the Domain Config Directory. This `config` directory can be found in the domain directory of a Sun Java System Application Server 8.1 instance, specifically the instance that will be protected by this agent. For example:

ApplicationServer-base/domains/domain1/config

where *ApplicationServer-base* represents the directory where the Sun Java System Application Server 8.1 software was installed.

The following is the default location for *ApplicationServer-base*:

/var/opt/SUNWappserver/

For deployments where instances are remote, see [Note on page 82](#) about the Domain Config Directory.

Instance Name: Enter the name of the Application Server instance that will be protected by the agent.

Admin Server Host: Enter the host on which the DAS for this instance is running.

Admin Server Host Is Remote: Enable this field only if the DAS for this instance is not on the host where the instance is running.

Admin Server Port: Enter the port number on which the DAS for this instance is available.

Admin Server Protocol: Select the appropriate protocol of the DAS. This protocol value may be either HTTP or HTTPS

Admin User Name: Enter the authorized domain Application Server administrative username

Admin Password: Enter the Administrator password for the domain. The password is defined as `AS_ADMIN_PASSWORD` in the password file.

Re-enter Admin Server Password: Re-enter the administrative password for the domain.

NOTE You must ensure that the Sun Java System Application Server 8.1 Domain Administration Server (DAS) is available at the time of installation of the agent.

For instances where DAS is not on the agent host, therefore DAS is remote, verify the DAS information is correct and the following field is enabled:

Admin server host is remote?

When DAS is remote, the domain config directory is available at the following location:

ApplicationServer-base/nodeagents/node1/instance1/config

where `instance1` is the Application Server instance the agent protects

and

where `node1` is the node agent to which `instance1` belongs.

For more information about node agents, see *Sun Java System Application Server 8.1 Administration Guide*.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the next section, the Summary Screen section.

Summary Screen

1. Click **Next** and in the Summary screen, review the information carefully. You can use the **Back** button to navigate back and change any information, if needed. Sensitive information like passwords are not displayed in this screen.
2. Once you have reviewed this information, click the **Next** button. The installation program now checks for the available disk space and displays the Ready to Install screen.

3. Click the Install Now button to start the installation of agent on your system.

Once you click the Install Now button, the installation program starts making changes to your system. When making these changes, the installation program will not allow you to cancel the installation.

The next screen displays the progress of installation as the installation program makes changes to your system. This screen does not require any user input or action and will automatically proceed to the next screen when the installation program finishes making the necessary changes to your system. This process may take a while. Take care not to terminate the process in between.

The next screen displays the installation summary. The installation program displays the status of the installation in this screen. Click on the Details button to see more information on the actions performed by the installation program.

NOTE The installation program for the IBM WebSphere Application Server 5.0/5.1 agent stops the application server at the end of the installation. You must manually restart it during post-installation.

At this stage, you are now ready to proceed to the [Post-Installation Tasks](#) section that details the tasks you must now perform in order to use the agent.

Using the Command-Line Installation Program

When the agent installation program is launched in the command line mode, it presents the user with a series of questions to gather the necessary information and report the status of the installation at certain stages of the overall installation. The command line installation program also provides active feedback, by way of text messages, to help the user enter correct information. The help messages for a set of installation questions are displayed before the common set of queries are made.

NOTE

- The installation program's command line interface expects a display terminal that has a minimum of 80 columns and 30 rows. It is also recommended that you have set your terminal buffer size to a considerably high number of rows to ensure that if some text were to scroll beyond the first row, you can use terminal's scroll bar to read it as necessary.
- At any time when the installation program prompts for information and it displays a value within brackets [] next to that prompt, you may select this value by simply pressing the `Enter` key. If you wish to enter a value other than the one displayed, you may type that value and then press the `Enter` key.

Follow these steps to install the agent in the command line mode:

1. Launch the installation program in the `nodisplay` mode as explained in the earlier sections. The installation program begins with a series of messages aimed at familiarizing you with the command line interface as well as providing some other necessary information. Read all these messages and press the `Enter` key on your keyboard to continue when prompted.

The installation program now displays the following message:

```
To install and configure this software, you must read and accept the entire
following Software Licence Agreement. After reading the agreement, you will
then be given the opportunity to accept or refuse the licensing terms.
Failing to accept the terms of the Software License Agreement will cause the
Installation program to end without installing the product.
```

```
<Press ENTER to display the Sun Software License Agreement>
```

2. At the prompt, press the `Enter` key to display the License agreement. The installation program now displays the License Agreement on the terminal.

NOTE Depending on your terminal size settings, it is possible that the License text may scroll too fast for you to read. In such a case, you must either use the scroll bar on your terminal to carefully read the License Agreement, or in case when using the scroll bar does not give access to the full License Agreement, you must read the accompanying `LICENSE.TXT` file for the terms and conditions of the License Agreement.

3. Read and understand all the terms and conditions as detailed in this agreement. If you do not agree with any term or condition of this agreement, enter `No` at the following prompt. If you agree to the terms and conditions, enter `Yes` to continue.

```
Have you read, and do you accept, all of the terms of the preceding Software
License Agreement [no] {"<" goes back, "!" exits}?
```

If you do not agree with these terms and conditions, the agent installation program will exit without making any changes to your system.

4. To install the agent in the displayed default directory, press only the `Enter` key. To use a different directory, type in the full path to the directory and press the `Enter` key.

```
Install Sun(tm) ONE Identity Server Policy Agent in this
directory [C:\Sun]{"<" goes back, "!" exits}:
```

5. At the following prompts, enter information about the Sun ONE Identity Server Services.

```
Enter the server information where the Sun ONE Identity Server
Service and Console are installed.

Sun ONE Identity Server Host [mycomputer.example.com] {"<" goes
back, "!" exits}
Sun ONE Identity Server Services Port [58080] {"<" goes back, "!"
exits}
Sun ONE Identity Server Services Protocol [http] {"<" goes back, "!"
exits}
Sun ONE Identity Server Services Deployment URI [/amserver] {"<"
goes back, "!" exits}
```

Sun ONE Identity Services Host: Enter the fully qualified host name of the system on which the Sun ONE Identity Server is installed.

Sun ONE Identity Server Services Port: Enter the port number for the Web Container that is used by Sun ONE Identity Server to provide its services.

Sun ONE Identity Server Services Protocol: Enter the appropriate protocol that will be used by the agent when communicating with Sun ONE Identity Server services. This protocol value may either be `HTTP` or `HTTPS`.

Sun ONE Identity Server Services Deployment URI: Enter the URI that will be used by the agent to access various Sun ONE Identity Server services.

6. At the following prompts, enter information about the Sun ONE Identity Server Console.

```
Sun ONE Identity Console Host [mycomputer.example.com] {"<" goes
back, "!" exits} :
Sun ONE Identity Server Console Port [58080] {"<" goes back, "!"
exits}
Sun ONE Identity Server Console Protocol [http] {"<" goes back,
"!" exits}:
Sun ONE Identity Server Console Deployment URI [/amconsole] {"<"
goes back, "!" exits}
amAdmin Password [] {"<" goes back, "!" exits}
Re-enter amadmin Password [] {"<" goes back, "!" exits}
amldapuser Password [] {"<" goes back, "!" exits}
Re-enter amldapuser Password [] {"<" goes back, "!" exits}
```

Sun ONE Identity Server Console Host: Enter the fully qualified hostname of the system on which the Sun ONE Identity Server console is installed.

Sun ONE Identity Server Console Port: Enter the port number of the Web Container that is used by Sun ONE Identity Server console.

Sun ONE Identity Server Console Protocol: Enter the appropriate protocol that will be used by the agent when communicating with Sun ONE Identity Server console. This protocol value may either be `HTTP` or `HTTPS`.

Sun ONE Identity Server Console Deployment URI: Enter the URI that will be used by the agent to access the Sun ONE Identity Server console.

amAdmin Password: Enter the amAdmin user password specified during Sun ONE Identity Server installation. The password entered here should be the amAdmin user password originally used during the installation of Sun ONE Identity Server. Even if after the installation of Sun ONE Identity Server, the password for the amAdmin user has been changed, you should still enter the original password and not the changed password.

Re-enter Password: Re-enter the amAdmin password to ensure that the correct value is used by the agent.

amldapuser Password: Enter the amldapuser password specified during Sun ONE Identity Server installation.

Re-enter amldapuser Password: Re-enter the amldapuser password to ensure that the correct value is used by the agent.

NOTE

- In situations where you are not using the features of the agent that require LDAP connectivity, you may choose to leave the amAdmin password blank. By doing so, the value for this password will be set to “changeit”, which can be changed at a later stage as necessary.
- If you select the HTTPS as either Sun ONE Identity Server Services Protocol, or Sun ONE Identity Server Console Protocol, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for the Application Server’s JDK keystore. Please refer to the Application Server documentation to learn how this can be done.

7. At the following prompts, provide details about the Directory Server corresponding to Sun ONE Identity Server services.

```
Enter the directory information corresponding to the Sun(tm) ONE
Identity Server Services
Directory Host [mycomputer.example.com] {"<" goes back, "!"
exits}
Directory Port [389] {"<" goes back, "!" exits}
Is the directory SSL enabled [false] {"<" goes back, "!" exits}?
Root Suffix [dc=iplanet,dc=com] {"<" goes back, "!" exits}
Installation Organization [dc=iplanet,dc=com] {"<" goes back, "!"
exits}
```

Directory Host: Enter the fully qualified hostname of the system where the Directory Server is installed.

Directory Port: Enter the port number used by this Directory Server.

SSL Enabled: Enter the value ‘true’ if the Directory Server uses SSL to communicate on the said port. Press the Enter key or enter ‘false’ if the Directory Server does not use SSL.

Root Suffix: Enter the root suffix to be used for accessing information stored in this Directory Server.

Installation Organization: Enter the complete DN of the default organization that was created when Sun ONE Identity Server was installed.

NOTE If you enter `true` for SSL Enabled prompt, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for the application server's JDK keystore. Please refer to the application server documentation to learn how this can be done.

8. At the following prompts, enter details of agent configuration.

```

Enter Agent configuration values.

Agent Host Name [mycomputer.example.com] {"<" goes back, "!"
exits}
Preferred protocol listening port [80] {"<" goes back, "!" exits}
Server's preferred protocol [http] {"<" goes back, "!" exits}
Config Reload Interval [10] {"<" goes back, "!" exits}
Enable Not-Enforced List Cache [false] {"<" goes back, "!" exits}
Number of Entries in Cache [1000] {"<" goes back, "!" exits}
Cache Expiration Time in Seconds [60] {"<" goes back, "!" exits}
Primary Application Context Path [/] {"<" goes back, "!" exits}
Enter Agent Filter Mode Options are: ALL, J2EE_POLICY,
URL_POLICY, SSO_ONLY, NONE [ALL] {"<" goes back, "!" exits}
Access Denied URL [] {"<" goes back, "!" exits}
Maximum allowed Login Attempts [5] {"<" goes back, "!" exits}

```

Agent Host Name: Enter the fully qualified hostname on which the application server protected by the agent is installed.

Preferred protocol listening port: Enter the preferred port number on which the application server provides its services.

Server's preferred protocol: Enter the protocol used by the application server to provide its services on the given port number as entered for the field Preferred protocol listening port. This protocol value may either be `HTTP` or `HTTPS`.

Config Reload Interval: Enter the amount of time in seconds after which the agent will automatically reload any changes made to the its configuration. Specify the value `0` to disable this feature.

The agent supports Hot-Swappable configuration, which can be changed without requiring the application server to be restarted. This feature can be very helpful in a controlled development and test environment where frequent changes to the configuration are needed to arrive to the correct configuration settings. It is recommended that this feature be disabled for production

systems to ensure the optimal use of system resources and to avoid accidental changes being picked up by the agent. Also note that although the majority of agent configuration is hot-swap enabled, there are some configuration settings which require a complete application server restart. Please refer to [Chapter 3, “Agent Configuration”](#) on page 153 to learn more about this feature.

Enable Not-Enforced List Cache: Enter `true` if you wish to enable the caching of Not-Enforced List entries as evaluated by the agent against incoming user requests.

It is recommended that the Not-Enforced List caching be enabled in order to optimize system performance. However, the overall system performance can degrade if the corresponding values of Cache Size and Cache Expiration time are not suited for your deployment. It is therefore recommended that the values used for these configuration settings be carefully evaluated in a controlled testing environment before being used in production. It should also be noted that the agent maintains two caches in its memory—one for recording the URIs that were evaluated as enforced and the other for recording the URIs that were evaluated as not-enforced. The specified values of Number of Entries in Cache and Cache Expiration time are equally applicable to both of these caches. This factor must be considered when setting the values for the size and expiration time of the cache.

Number of Entries in Cache: Enter the size of cache used by the agent to store the evaluated results.

Cache Expiration Time in Seconds: Enter the amount of time in seconds after which the agent can purge an entry from its cache to free up the cache memory for newer entries.

Primary Application Context Path: Enter the Context path for the primary application being protected by this agent. If this application is deployed at the root context of the application server, enter the value “/”.

When installing the agent for SAP Enterprise Portal 6.0 SP2, enter the context path as `/irj`.

When installing the agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, you should ensure that the primary application context path is set to the context path of the application that will be available throughout the life of the server. If your SAP installation does not include the

Enterprise Portal, you must specify the value corresponding to one of the applications that will be protected by this agent. If, however, your SAP installation does include the Enterprise Portal, you may use the context path `/irj` as well

In a situation where there are multiple applications deployed on the application server, you must choose the context path of the application that is available for as long as the application server is alive. This value is used by the agent to receive notifications from Sun ONE Identity Server and is also used to provide support for legacy browsers, etc. Providing an incorrect value for this configuration setting may lead to the malfunction of the agent and the application server.

Agent Filter Mode: Enter the agent filter mode that you would like to run the agent in. For a complete description of the available agent filter modes, refer to the chapter [Chapter 3, “Agent Configuration” on page 153](#).

When installing the agent for SAP Enterprise Portal 6.0 SP2, the agent filter mode must be set to either `SSO_ONLY` or `URL_POLICY`. Other agent filter modes are not supported by this agent.

Access Denied URL: Enter the complete URI for the access denied page to be used by the agent. If this value is not specified, the agent will use `HTTP` error status code `403 (FORBIDDEN)` to deny access to resources as needed.

Maximum allowed Login Attempts: Enter the number of unsuccessful access attempts in succession after which the user will not be allowed to access the requested URL temporarily for security purposes. Specify the value `0` to disable this feature.

This feature can be used to guard the hosted application from Denial-Of-Service attacks where the end user can overload the application server by repeated authentication requests. By disabling this feature, the system remains vulnerable to such attacks. Therefore this feature should not be disabled unless there is a specific requirement that necessitates the disabling of this feature.

NOTE If you select `HTTPS` as Server's preferred protocol, you must ensure that the Certificate Authority certificate for the signer of the corresponding server certificates is added to the trusted list for Sun ONE Identity Server. Please refer to Sun ONE Identity Server documentation to learn how this can be done.

9. At the following prompts, provide information about the JDK installation directory for the application server and JCE/JSSE configuration for that specific JDK:

```

Enter the server information where the JDK is installed.
JDK Directory [C:\jdk1.3.1_01] {"<" goes back, "!" exits}
Install JCE for this JDK [false] {"<" goes back, "!" exits}
Install JSSE for this JDK [false] {"<" goes back, "!" exits}

```

JDK Directory: Enter the complete path to the directory where the JDK being used by your application server is installed.

Install JCE for this JDK: Enter the value `true` if the JDK being used by your application server does not have JCE installed.

Install JSSE for this JDK: Enter the value `true` if the JDK being used by your application server does not have JSSE installed.

NOTE

- If the JDK being used by your application server does not have JCE installed, you must select the Install JCE for this JDK checkbox. Failing to do so can result in the installation program failure.
 - If your application server is based on JDK 1.4 or above, the options to install JCE and JSSE will not be provided by the agent installation program.
 - If you are installing the agent for BEA WebLogic Server 8.1, you must accept the default values (`false`) for the prompts Install JCE for this JDK and Install JSSE for this JDK.
-

- 10.** If you are installing the agent for Tomcat Server 4.1.27, Oracle 9iAS R2, Oracle 10g, SAP Enterprise Portal 6.0 SP2, or Macromedia JRun 4, the installation program will now display the following prompts where you must enter the user and group information for the process owner of the application server.

```

Enter the user and group information for the process owner of the
Application Server.

```

```

User Name [root] {"<" goes back, "!" exits} root
Group Name [root] {"<" goes back, "!" exits} other

```

User Name: The user name of the user running the application server process.

Group Name: The group name of the group to which the user running the Application Server process belongs to.

NOTE

- The above prompts are displayed only when the agent is installed on various Unix platforms.
 - Valid but inappropriate values supplied for the user name and/or group name can lead to severe problems with the configuration of the application server and can render the application server unusable.
 - The server instance should be started or stopped only by the process owner of the application server. If these tasks are done by another user, it will cause the agent to malfunction.
-

The prompts displayed thus far by the installation program are the same for all the agents. However, the prompts that follow are specific to the application server agent that is being installed. From the following list, click the appropriate link depending on which application server agent you are installing. After you perform the steps in the respective section, proceed to the [Summary Information Prompt](#) section.

- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic Server 6.1 SP2](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1](#)
- [Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

If you are installing the agent for Sun ONE Application Server 7.0, enter the configuration information at these prompts as explained in this section.

```

Enter the answers to Sun ONE Application Server specific questions as
prompted by the agent installation program:
Enter Sun ONE Appserver configuration Details:
Application Server bin directory [C:\Sun\AppServer7\bin] {"<"
goes back, "!" exits}
Instance Config Directory
[C:\Sun\AppServer7\domains\domain1\server1\config] {"<" goes
back, "!" exits}
Admin User Name [admin] {"<" goes back, "!" exits}
Admin Server Port [4848] {"<" goes back, "!" exits}
Admin Server Password [] {"<" goes back, "!" exits}
Re-enter Admin Server Password [] {"<" goes back, "!" exits}

```

Application Server bin directory: Enter the complete path to the `bin` directory of the Sun ONE Application Server 7.0 installation.

Instance Config Directory: Enter the complete path to the `config` directory of the Sun ONE Application Server 7.0 instance which will be protected by this agent.

Admin User Name: Enter the user name of the Administrator of this Sun ONE Application Server 7.0 instance.

Admin Server Port: Enter the port number on which the Administration Server for this Sun ONE Application Server 7.0 is available.

Admin Password: Enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

Re-enter Admin Server Password: Re-enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

NOTE You must ensure that the Sun ONE Application Server 7.0 Administration Server is running at the time of the agent installation.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for BEA WebLogic Server 6.1 SP2

If you are installing the agent for BEA WebLogic Server 6.1 SP2, answer the following prompt as explained here:

```
Startup Script Location
[C:\bea\wlserver6.1\config\mydomain\startWebLogic.cmd] {"<" goes
back, "!" exits}
```

Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server instance.

-
- NOTE**
- When the WebLogic Server is in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the CLASSPATH and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of CLASSPATH and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent CLASSPATH to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for IBM WebSphere Application Server 5.0/5.1

If you are installing the agent for IBM WebSphere Application Server 5.0/5.1, enter the following information at these prompts.

```
Enter WebSphere Application Server configuration Details:
WebSphere Root Directory [/space/ibm/WebSphere/AppServer] {"<"
goes back, "!" exits}
Instance Config Directory
[/space/ibm/WebSphere/AppServer/config/cells/heros/nodes/euros/s
ervers/server1] {"<" goes back, "!" exits}
WebSphere Admin Connector [SOAP] {"<" goes back, "!" exits}
WebSphere Admin Connector Port [8880] {"<" goes back, "!" exits}
WebSphere Cell Name [euros] {"<" goes back, "!" exits}
WebSphere Node Name [euros] {"<" goes back, "!" exits}
WebSphere Server Instance Name [server1] {"<" goes back, "!"
exits}
```


WebSphere Root Directory: Enter the complete path to the root directory for the WebSphere Application Server.

Instance Config Directory: Enter the complete path to the config directory of the WebSphere Application Server instance that will be protected by this agent.

WebSphere Admin Connector: Specify the protocol for connecting to WebSphere Administration Service. You can choose between `SOAP` and `RMI`.

WebSphere Admin Connector Port: Enter the Administration Service Port for specified Connector.

WebSphere Cell Name: Enter the cell name for the IBM WebSphere Application Server.

WebSphere Node Name: Enter the node name for the IBM WebSphere Application Server.

WebSphere Server Instance Name: Enter the instance name of the IBM WebSphere Application Server instance.

NOTE

- You must ensure that the IBM WebSphere Application Server 5.0/5.1 instance is running at the time of the agent installation.
 - The agent installation program stops the IBM WebSphere Application Server at the end of the installation. You may manually restart it later.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1

If you are installing the agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1, answer the following prompts as explained here:

```
Startup Script Location
[/bea/user_projects/mydomain/startWebLogic.sh] {"<"goes back, "!"
exits}
WebLogic Server MBeans directory
[/bea/weblogic700/server/lib/mbeantypes] {"<" goes back, "!"
exits}
```

Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server instance. This script is typically located under the following directory:

WebLogic_Install_Dir/boa/user_projects/server-domain-name

WebLogic Server MBeans Directory: Enter the full path to the `mbeantypes` directory:

WebLogic_Install_Dir/boa/weblogic700/server/lib/mbeantypes

This directory stores MBeans required to deploy on WebLogic.

NOTE

- When the WebLogic Server is being operated in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the `CLASSPATH` and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of `CLASSPATH` and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent `CLASSPATH` to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for PeopleSoft 8.3/8.4/8.8

If you are installing the agent for PeopleSoft, perform the following steps:

1. At the following prompt, enter the item number to choose the PeopleSoft application version on which you are going to install the agent.

```
Choose the PeopleSoft version on which you want to install the agent:
```

```
PeopleSoft HRMS 8.3
```

```
PeopleSoft Financial 8.4
```

```
PeopleSoft HRMS 8.8
```

```
Choose one of the options from above [1] {"<" goes back, "!" exits}
```

2. Enter the item number to choose one of the following agent deployment options.

```
1. Agent on WebLogic Server hosting PeopleSoft application
2. Proxy Solution - Agent on a Separate Proxying Web Server

Choose one of the options from above [1] {"<" goes back, "!" exits}
```

Agent on WebLogic Server hosting PeopleSoft application: Select this option to install the agent on top of BEA WebLogic Server hosting the PeopleSoft application.

Proxy Solution - Agent on a Separate Proxying Web Server: Select this option if you want to configure Sun ONE Web Server (also referred as iPlanet Web Server) as a proxy for BEA WebLogic Server that hosts the PeopleSoft application.

NOTE

When installing the agent on PeopleSoft 8.3 HRMS with BEA WebLogic 5.1 front-end server, ensure that the BEA WebLogic 5.1 server has the Service Pack 12 installed. Not having this service pack installed on BEA WebLogic server could result in the malfunction of the PeopleSoft application after the installation of the agent.

To upgrade the BEA WebLogic server to Service Pack 12, you can download the binaries from PeopleSoft distribution site located at <ftp://ftp.peoplesoft.com/outgoing/GSC/jim/weblogic510>. Please follow the instructions provided at this location to upgrade the server.

3. At the following prompts, provide the PeopleSoft installation information.

```
PeopleSoft Installation Details

PeopleSoft Application Server installed locally [true] {"<" goes back,
"!"exits}
WebLogic Server installed locally [true] {"<" goes back, "!" exits}
PeopleSoft AppServer Directory [/usr/psft/appserv] {"<" goes back, "!"exits}
PeopleSoft Domain Name [HDMO] {"<" goes back, "!" exits}
WebLogic Instance Directory [/usr/psft/weblogic/myserver] {"<" goes back, "!"
exits}
PeopleSoft user ID [psft] {"<" goes back, "!" exits}
PeopleSoft user's primary group [sys] {"<" goes back, "!" exits}
```

PeopleSoft Application Server Installed locally: Press Enter to specify that you have PeopleSoft Application Server installed locally.

WebLogic Server Installed locally: Press Enter if you have WebLogic Server installed locally.

NOTE Select one or both of the above options according to your installation. You need to select at least one to proceed.

PeopleSoft AppServer Directory: Enter the full path to the directory where the PeopleSoft Application Server is installed. For example, *PS_HOME/appserv*

PeopleSoft Domain Name: Enter the PeopleSoft domain that this agent will protect. For example, *HDMO*.

WebLogic Instance Directory: Enter the full path to the WebLogic instance directory. For example, *PS_HOME/weblogic/myserver* or *BEA_HOME/wlserver6.1/config/peoplesoft*

PeopleSoft user ID: Enter the system user ID used to install PeopleSoft software. For example, *psft*

PeopleSoft user's primary group: Enter the primary group of the PeopleSoft user. For example, *sys*

4. At the following prompts, enter the locations of the application server's JDK.

```
Enter the server information where the JDK is installed.
Directory containing JRE used by PeopleSoft [/export/home/psft4/pt] {"<" goes
back, "!" exits}
Install JCE for this [true] {"<" goes back, "!" exits}
Install JSSE for this [true] {"<" goes back, "!" exits}
WebLogic JDK Directory [/export/home/psft4/bea/jdk131] {"<" goes back,
"!"exits}
Install JCE for this [true] {"<" goes back, "!" exits}
Install JSSE for this [true] {"<" goes back, "!" exits}
```

Directory containing JRE used by PeopleSoft: Enter the complete path to the directory containing the JRE used by the application server.

Install JCE for this: Press Enter if the Sun JCE provider needs to be installed on the JDK.

Install JSSE for this: Press Enter if the Sun JSSE provider needs to be installed on the JDK.

WebLogic JDK Directory: Enter the complete path to the JDK used by WebLogic Server.

Install JCE for this: Press Enter if the Sun JCE provider needs to be installed on the JDK.

Install JSSE for this: Press Enter if the Sun JSSE provider needs to be installed on the JDK.

5. At the following prompts, enter the configuration details for this installation.

Configuration details for this installation.

```
PeopleSoft DEFAULT_USER [DEFAULT_USER] {"<" goes back, "!" exits}:
PeopleSoft DEFAULT_USER Password [] {"<" goes back, "!" exits}:
Re-enter Password [] {"<" goes back, "!" exits}:
Name of the PIA Site [ps] {"<" goes back, "!" exits}:
Name of the PIA web application [PORTAL] {"<" goes back, "!" exits}:
```

PeopleSoft DEFAULT_USER: This field displays the generic PeopleSoft user ID that the web server uses to identify itself to the application server. For example, DEFAULT_USER.

PeopleSoft DEFAULT_USER Password: Enter the PeopleSoft DEFAULT_USER password assigned while creating the DEFAULT_USER.

Re-enter Password: Enter the password for DEFAULT_USER again for confirmation.

Name of the PIA Site: Enter the name of the PIA site, as given during PeopleSoft installation. For example, peoplesoft8.

Name of the PIA web application: Enter the name of the PIA web application as given during PeopleSoft installation. For example, PORTAL.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for Apache Tomcat Server 4.1.27

If you are installing the agent for Tomcat Server 4.1.27, the installation program will now display the following prompts.

```
Enter Tomcat Server Configuration Information

Tomcat server instance configuration directory
[/opt/jakarta-tomcat-4.1.27/conf] {"<" goes back, "!" exits}:
Install agent filter in global web.xml [true] {"<" goes back, "!" exits}?
```

Instance Configuration Directory: Provide the complete path to the configuration directory of the Tomcat Server Instance.

Enable Global Filter: Enter `true` if you want to install the agent filter for the global `web.xml`, otherwise enter `false`. Enabling global filter will enforce Identity Server policies on the entire instance including its index page, if any. Moreover, you will not be required to manually add filter definition for any individual web application deployed on this instance.

NOTE There is more than one way to deploy the agent filter. To see details about both these options, see the section [Agent for Apache Tomcat Server 4.1.27](#) under [Post-Installation Tasks](#).

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for Oracle 9iAS R2 and Oracle 10g

When installing the agent for Oracle 9iAS R2 or Oracle 10g, you must enter the following information about the Oracle Server that is to be secured by the agent.

```
Oracle instance config directory [/ora/j2ee/home/config] {"<" goes back, "!"
exits} /export/home/orcl/ora9ias/j2ee/home/config
```

Oracle instance config directory: Enter the complete path to the Oracle instance configuration directory. Ensure that you have `write` privileges for this directory.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for SAP Enterprise Portal 6.0 SP2

If you are installing the agent for SAP Enterprise Portal 6.0 SP2, enter the configuration information on this screen as explained in this section.

```
Enter SAP Enterprise Portal Information

Server Directory Location [/usr/sap/porj2ee1/j2ee/j2ee_01/cluster/server]
{"<" goes back, "!" exits}
```

Server Directory Location: Enter the complete path to the directory that contains the server installation to be protected by this agent.

NOTE You must ensure that the SAP Enterprise Portal server is not running during the agent installation.

Agent for Macromedia JRun 4

If you are installing the agent for Macromedia JRun 4, enter the configuration information at these prompts as explained in this section.

```
Enter JRun Server Configuration Information

JRun server instance configuration directory
[/opt/jrun4/servers/default/SERVER-INF] {"<" goes back, "!" exits}:

Install agent filter in default-web.xml [true] {"<" goes back, "!" exits}?
```

Instance Configuration Directory: Enter the complete path to the configuration directory of the Macromedia JRun Server Instance.

Install agent filter in default-web.xml: Enter `true` if you want to install the agent filter for `default-web.xml`, otherwise enter `false`. If you choose `true`, the agent will enforce Sun ONE Identity Server policies on the specified JRun server instance, including its index page, if any. However, you will not be required to manually add filter definition to every individual web application deployed on this instance anymore.

NOTE If you select this option, all resources in the server instance will be protected by the servlet filter defined in the `default-web.xml` and users will not be able to access even the welcome page of the server, if any, without providing the credentials and adding a rule to access that resource in Sun ONE Identity Server. One way to handle this problem is to add your server's welcome page URL or any other resource that does not need the protection, to the property `com.sun.am.policy.amFilter.notenforcedList[<index>]` in the agent configuration file `AMAgent.properties`.

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

If you are installing the agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, enter the configuration information on this screen as explained in this section.

Enter SAP Server Information

Server Directory Location [D:\usr\sap\TNG\DVEBMGS00\j2ee\cluster\server]
{ "<" goes back, "!" exits }

Server Directory Location: Enter the complete path to the installation directory of the server that will be protected by this agent. Use the browse button to locate this directory if necessary.

NOTE Ensure that the SAP Server is not running during the agent installation.

Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal

If you are installing the agent for WebLogic 8.1 Server/Portal, answer the following prompts as explained in this section:


```
Startup Script Location [/bea/user_projects/mydomain/startWebLogic.sh]
{"<"goes back, "!" exits}

WebLogic Server MBeans directory [/bea/weblogic/server/lib/mbeantypes]{"<"
goes back, "!" exits}
```

Startup Script Location: Enter the full path to the startup script that is used to start your BEA WebLogic Server/Portal instance. This script is typically located under the following directory:

WebLogic_Portal_Install_Dir/user_projects/portal-domain-name

WebLogic Server MBeans Directory: Enter the full path to the `mbeantypes` directory:

WebLogic_Portal_Install_Dir/weblogic81/server/lib/mbeantypes

This directory stores MBeans required to deploy on WebLogic.

NOTE

- When the WebLogic Portal/Server is in a cluster environment, the agent installation program may not update the startup script used to launch a managed server. In this situation, the administrator must manually edit the script and insert all the `CLASSPATH` and Java VM options that were inserted in the original startup script by the installation program. For the exact details on the values of `CLASSPATH` and Java VM options, please refer to the startup script that was modified by the agent installation program.
 - When modifying any startup script to add the agent `CLASSPATH` to the Java VM, you must ensure that the agent classes are placed before the WebLogic Server classes in the final class path. Failing to do so can cause the agent to malfunction during runtime.
-

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the [Summary Information Prompt](#) section.

Agent for Sun Java System Application Server 8.1

If you are installing the agent for Sun Java System Application Server 8.1, answer the following prompts as explained in this section:

```

Enter Sun ONE Appserver configuratioin Details
Application Server bin directory [/opt/SUNWappserver/appserver/bin] {"<"
  goes back, "!" exits}
Domain Config Directory[/var/opt/SUNWappserver/domains/
  domain1/config]{"<" goes back, "!" exits}
Instance Name [server] {"<" goes back, "!" exits}
Admin Server Host [example.company22.com] {"<" goes back, "!" exits}
Admin Server Host is Remote ? [false] {"<" goes back, "!" exits}
Admin Server Port [4849] {"<" goes back, "!" exits}
Admin User Name [admin] {"<" goes back, "!" exits}
Admin Server Password [] {"<" goes back, "!" exits}
Re-enter Admin Server Password [] {"<" goes back, "!" exits}

```

Application Server Bin Directory: Enter the complete path to the bin directory of the Sun Java System Application Server 8.1 installation.

Domain Config Directory: Enter the complete path to the Domain Config Directory. This config directory can be found in the domain directory of a Sun Java System Application Server 8.1 instance, specifically the instance that will be protected by this agent. For example:

ApplicationServer-base/domains/domain1/config

where *ApplicationServer-base* represents the directory where the Sun Java System Application Server 8.1 software was installed.

The following is the default location for *ApplicationServer-base*:

/var/opt/SUNWappserver/

For deployments where instances are remote, see [Note on page 105](#) about the Domain Config Directory.

Instance Name: Enter the name of the Application Server instance that will be protected by the agent.

Admin Server Host: Enter the host on which the DAS for this instance is running.

Admin Server Host Is Remote: Enable this field only if the DAS for this instance is not on the host where the instance is running.

Admin Server Port: Enter the port number on which the DAS for this instance is available.

Admin Server Protocol: Select the appropriate protocol of the DAS. This protocol value may be either HTTP or HTTPS.

Admin User Name: Enter the authorized domain Application Server administrative user name.

Admin Password: Enter the Administrator password for the domain. The password is defined as `AS_ADMIN_PASSWORD` in the password file.

Re-enter Admin Server Password: Re-enter the administrative password for the domain.

NOTE You must ensure that the Sun Java System Application Server 8.1 Domain Administration Server (DAS) is available at the time of installation of the agent.

For instances where DAS is not on the agent host, therefore DAS is remote, verify the DAS information is correct and the following field is enabled:

Admin server host is remote?

When DAS is remote, the domain config directory is available at the following location

ApplicationServer-base/nodeagents/node1/instance1/config

where `instance1` is the Application Server instance the agent protects

and

where `node1` is the node agent to which `instance1` belongs.

For more information about node agents, see *Sun Java System Application Server 8.1 Administration Guide*.

Next, the installation program displays a summary of all the information that you have provided to install the agent. For details about this screen and to get started with the installation, proceed to the next section, the [Summary Information Prompt](#) section.

Summary Information Prompt

1. At the next prompt, review the summary of information as gathered by the installation program. You can use the “<” option to navigate back and change any information, if needed. Sensitive information like passwords are not displayed in the console. To continue press the `Enter` key on your keyboard.

The installation program now checks for the necessary disk space and displays a Ready to Install prompt.

2. Choose from the given choices to proceed. You may:
 - continue with the installation by selecting the option 1,
 - start over again by using the option 2, or
 - simply exit the installation program by using the option 3.

If you choose to continue, the installation program will start making changes to your system. If you choose to exit, no changes will be made to your system.

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

On continuing with the installation, the installation program displays the progress of the installation as changes are made to your system. This phase does not require any input and the installation program proceeds to the next prompts when installation changes being made to your system are complete. It may take a while to display the next prompt. Take care not to terminate the process.

```
Progress
| -1%-----25%-----50%-----75%-----100% |
```

The final prompt displayed by the installation program provides the status of the installation and gives you an option to either view the details of the installation logs, or simply exit the installation program.

```

Installation Details:
Product                               Result                               More Information
1. Sun ONE Identity Server Policy Agent  Installed                           Available
2. Done
Enter the number corresponding to the desired selection for more
information, or enter 2 to continue [2] {"!" exits}:
To view installation log enter the choice 1. To exit the installation program, enter 2.

```

At this stage, you are now ready to proceed to the next section that details the post-installation tasks that must be performed in order to use the agent.

NOTE The installation program for the IBM WebSphere Application Server 5.0/5.1 agent stops the application server at the end of the installation. You must manually restart it during post-installation.

Post-Installation Tasks

Once the agent is installed, you must follow the steps in this section to configure the agent to run with your application server and the deployed and protected applications. You can directly go to the agent-specific section from the list given below to see the tasks you need to perform in order to enable the agent for your application server.

- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic Server 6.1 SP2](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1](#)
- [Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)

- [Agent for Sun Java System Application Server 8.1](#)
- [Common Tasks](#)

Agent for Sun ONE Application Server 7.0

Once you have installed the agent for Sun ONE Application Server 7.0, you must complete the following post-installation task:

- [Install the Agent Filter for the Deployed Application](#)

The following is a detailed description of this task:

Install the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of sub-tasks explains how the agent filter can be installed for the application you want to be protected by the agent.

1. To install the agent filter, you must ensure that the application is not currently deployed on the application server. If it is currently deployed, you must remove it before proceeding any further.
2. In order to install the agent filter, you will have to modify the deployed application's deployment descriptor. It is therefore recommended that you create the necessary backups before proceeding to modify these descriptors.
3. Edit the application's `web.xml` descriptor as follows:

Since filters were introduced in [Servlet Specification 2.3](#), the `web.xml`'s `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

After modifying the `<DOCTYPE>` element, you now need to add the `<filter>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and the `<filter-mapping>` elements immediately following the description element of the `<web-app>` element in the descriptor `web.xml`. The following is a sample `web.xml` descriptor with the `<filter>` and the `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun™ ONE Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the `web.xml` deployment descriptor is modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter is added to the application. You can now redeploy your application on the application server.

NOTE The J2EE agent `filter-class` attribute is backward compatible. So, an application protected by a previous version of agent filter class (for example `com.sun.amagent.as.filter.AgentFilter`) does not need to modify `web.xml` to reflect the new filter class for the latest version of the agent.

Agent for BEA WebLogic Server 6.1 SP2

Once you have installed the agent for BEA WebLogic Server 6.1 SP2, you must complete the following post-installation tasks:

- [Installing the Agent Realm for the BEA WebLogic Server 6.1 SP2](#)
- [Installing the Agent Filter for the Deployed Application](#)

Installing the Agent Realm for the BEA WebLogic Server 6.1 SP2

Once the agent has been installed on your system, the BEA WebLogic Server must be configured to use the Realm provided as a part of the agent installation. The following list of sub-tasks explains how the agent realm can be installed for your BEA WebLogic Server installation.

- [Creating a Custom Realm for the Agent](#)
- [Creating a Caching Realm for Agent Realm](#)
- [Configuring the File Realm](#)

NOTE This section outlines the steps necessary to successfully add the agent realm to the WebLogic Server. It must be noted that the information provided in this section is only to facilitate the installation of agent realm and should not be taken as a substitute for the information provided in WebLogic Server documents. For a complete in-depth discussion on WebLogic Custom Realms, refer WebLogic Server documentation at: <http://www.bea.com>.

Creating a Custom Realm for the Agent

NOTE The WebLogic Realm class name attribute is backward compatible. So if the realm has been previously installed using the Policy Agent, version 1.0 or 2.0, it will continue to function appropriately. If so, you can skip this subtask and directly jump to the next sub-task.

1. Make sure that the BEA WebLogic Server is currently running.
2. In the left pane of the WebLogic Server Administration Console, expand the Security Node by clicking the “+” sign.
3. In the left pane under Security node, click the “Realms” item to display a list of available Realms in the system on the right pane.
4. On the right pane, click the “Configure a new Custom Realm” link. This displays a form that can be used to enter the information regarding the new Custom Realm that you are trying to create.

5. In this form, enter the following information and click on Create:

Name: Agent Realm

Realm Class Name: com.sun.identity.agents.weblogic.AmWebLogicRealm

6. After creating the new Realm, restart the WebLogic Server.

Once the WebLogic Server is restarted, using the Administration Console, navigate to Security > Realms node. You should be able to see the newly created agent realm in the list of Realms displayed on the right pane.

Creating a Caching Realm for Agent Realm

1. Logon to the WebLogic Server Administration Console. Use the configured system username and password for logging on to the console.
2. In the left pane of the WebLogic Server Administration Console, expand the Security Node by clicking the “+” sign next to it.
3. In the left pane under Security node, click the Caching Realms to display a list of available Caching Realms in the system on the right pane.
4. On the right pane, click the “Configure a new Caching Realm” link. This displays a form that can be used to enter the information regarding the new Caching Realm that you are trying to create.
5. In this form, enter the following information:

Name: Agent Caching Realm

Basic Realm: Select Agent Realm from the pull down menu.

6. Click on the Create button. This refreshes the right pane and a new Caching Realm is created. The right pane displays the configuration of this newly created Caching Realm.
7. In the right pane, disable all the caching attributes. Click the ACL Tab. This displays the ACL caching attributes. Uncheck the check box next to Enable ACL Cache and Click on the Apply button.
8. Repeat the last step for the rest of the available tabs—Authentication, Groups, Users, and Permissions. Uncheck the appropriate Enable Cache check box and click the Apply button.

9. Restart the WebLogic Server.

Once the WebLogic Server is restarted, using the Administration Console, navigate to Security > Caching Realms node. You should be able to see the newly created Agent Caching Realm in the list of Caching Realms displayed on the right pane.

Configuring the File Realm

1. Logon to the WebLogic Server Administration Console.
2. On the left pane click the Security node. This forces the console to display the Security configuration of the WebLogic Server in the right pane.
3. In the right pane, click the Filerealm tab. This causes the console to display the details of the current File Realm.
4. In the form that is displayed on the right pane, under Caching Realm, select the Agent Caching Realm from the pull down menu and click on the Apply button.
5. Restart the WebLogic Server.

Once the File Realm is configured and the WebLogic Server is restarted, the agent realm has been successfully installed.

NOTE It is recommended that after the agent realm has been successfully configured, you should create a backup of the WebLogic Server's `config.xml` file. You may name this backup as `config.xml.withAgent`, so that the next time you install the agent, you can simply copy this file on top of your existing `config.xml` and bypass the manual steps necessary to install the agent realm.

Installing the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of sub-tasks detail how the agent filter can be installed for your application that must be protected by the agent.

1. To install the agent filter you must ensure that the application is not currently deployed on the application server. If it is currently deployed you must remove it before proceeding any further.
2. In order to install the agent filter, you will have to modify the deployed application's deployment descriptor. It is therefore recommended that you create the necessary backups before proceeding to modify these descriptors.

3. Edit the application's `web.xml` descriptor as follows:

Since filters were introduced in [Servlet Specification 2.3](#), the `web.xml`'s `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as follows:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

After modifying the `<DOCTYPE>` element, you now need to add the `<filter>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and the `<filter-mapping>` elements immediately following the description element of the `<web-app>` element in the descriptor `web.xml`. The following is a sample `web.xml` descriptor with the `<filter>` and the `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun™ ONE Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the `web.xml` deployment descriptor is modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter is added to the application. You can now redeploy your application on the application server.

NOTE The J2EE agent filter-class attribute is backward compatible. So, an application protected by a previous version of agent filter class (for example `com.sun.amagent.as.filter.AgentFilter`) does not need to modify `web.xml` to reflect the new filter class for the latest version of the agent.

Agent for IBM WebSphere Application Server 5.0/5.1

Once you have installed the agent for IBM WebSphere Application Server 5.0/5.1, you must complete the following post-installation tasks:

- [Enabling Security for IBM WebSphere Application Server](#)
- [Installing the Agent Filter for the Deployed Application](#)

Enabling Security for IBM WebSphere Application Server

1. Start the Application Server instance, for which the agent was configured as part of the installation.
2. Login to the IBM WebSphere Administrative Console.
3. Enable Global Security. This is under the Global Security Settings.
4. Optionally enable Java 2 Security, if your applications are designed to work with Java 2 Security enabled.
5. Apply the changes and save the changes to Master configuration.
6. Perform this step when the following two conditions apply:
 - When configuring the agent against an SSL enabled Identity Server
 - If the IBM WebSphere Application Server was installed with samples

Remove the sample URL provider for the HTTPS protocol installed as a part of samples. This sample URL provider conflicts with the https URL provider (from IBM JSSE implementation) used by the agent. Search for and, if found, remove this provider as follows:

- a. Log on to the WebSphere Application Server Administration Console.
- b. In the left panel, click the Resources node.

This expands the node to reveal various resources that can be configured for the server.

- c. Click the URL Providers resource link.

This loads the corresponding properties on the right content panel.

- d. In the right content panel, expand the Scope node by clicking it, if it is not already expanded.
- e. Select the radio button next to the server entry.

- f. Click the Apply button.
- g. Search the list of properties displayed at the bottom of the panel for the Samples URL Provider - https entry.
- h. If this provider is present, select the check box next to it.
- i. Click Delete.
- j. Click Save at the top of the page to save this change to the master configuration.

This brings you to a page that has the Save button.

- k. Click Save to commit your changes.

7. Restart the Application Server.

NOTE Once the global security is enabled and the IBM WebSphere application Server restarted, the WebSphere administration console becomes password protected. The only user that is allowed to access this console is the `amldapuser`. Therefore in order to access the console for any administration purposes, you must supply the user name `amldapuser` and the corresponding password before the agent grants access to the WebSphere Administration console.

Installing the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of sub-tasks detail how the agent filter can be installed for your application that must be protected by the agent.

1. To install the agent filter you must ensure that the application is not currently deployed on the application server. If it is currently deployed you must remove it before proceeding any further.
2. In order to install the agent filter, you will have to modify the deployed application's deployment descriptor. It is therefore recommended that you create the necessary backups before proceeding to modify these descriptors.
3. Edit the application's `web.xml` descriptor as follows:

Since filters were introduced in [Servlet Specification 2.3](#), the `web.xml`'s `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

After modifying the `<DOCTYPE>` element, you now need to add the `<filter>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and the `<filter-mapping>` elements immediately following the description element of the `<web-app>` element in the descriptor `web.xml`. The following is a sample `web.xml` descriptor with the `<filter>` and the `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun™ ONE Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.websphere.AmWAS50AgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the `web.xml` deployment descriptor has been modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter has been added to the application. You can now redeploy your application on the application server.

Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1

Once the policy agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1 has been installed on your system, you must configure the WebLogic Server to use the agent authentication provider as explained in the following sections:

- [Configuring the Agent Authentication Provider](#)

- [Application Configuration](#)
- [Installing the Agent Filter in an Application](#)

NOTE The agent can be installed irrespective of whether the server is running or not. But if the server was running during the installation, it needs to be restarted before configuration.

Configuring the Agent Authentication Provider

The agent authentication provider can be added to the WebLogic Server by using the WebLogic Server Console. This section outlines the steps necessary to add the agent authentication provider to the WebLogic Server. It must be noted that the information provided in this section is only to facilitate the configuration of agent authentication provider and should not be taken as a substitute for the information provided in WebLogic Server documents. For a complete in-depth discussion on WebLogic Authenticators, refer WebLogic Server documentation located at <http://www.bea.com>.

To configure the server to use the agent authentication provider:

1. Logon to the WebLogic Server console.
2. On the left frame, click on *agentdomain* > Security > Realms > myrealm, where *agentdomain* is the domain you had configured. It is not mandatory that you select the default 'myrealm.' You may configure a new realm and select it for configuration.
3. On the right frame, click on the Providers tab.
4. Click on Authentication.
5. Click on the link Configure a New Agent Authenticator.
6. Click Create to create a new agent authenticator.
7. Change the control flag value from REQUIRED to OPTIONAL and click Apply.
8. Go back to Agent Providers tab and you should now be able to see an Agent Authenticator.
9. Now click on Default Authenticator.
10. Change the control flag from REQUIRED to OPTIONAL. Click Apply.

NOTE It is not mandatory that you use the default 'myrealm' to configure the agent. You can also use a new realm to configure the agent. However, in that case, you must make sure that the control flag value for the Agent Authenticator is set to OPTIONAL. Also make sure that any additional Authentication Providers you use are also set to OPTIONAL.

11. Restart WebLogic Server.

Application Configuration

The agent authentication provider component of the policy agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1 provides runtime mapping of various principals in Sun ONE Identity Server. Abstract security role names are used by the hosted application in order to determine if the currently authenticated user is authorized to access a particular resource or is otherwise a member of a given role. This runtime evaluation can occur only if the user is authenticated as a Sun ONE Identity Server principal by the means of Sun ONE Identity Server's authentication service. Without the user being authenticated appropriately, the results of such evaluations done by the agent authentication provider will always be negative, resulting in access being denied to the user for the requested resource.

It is the agent filter component that enforces authentication for users who try to access particular application resources, thereby enabling the agent authentication provider to correctly evaluate the principal mappings as desired.

Unlike the agent authentication provider, which is installed in the core of WebLogic Server, the agent filter is installed in the deployed application which must be protected by Sun ONE Identity Server. This is true for every application that must be protected on the WebLogic Server using the agent. It is recommended that applications that are not protected using the agent should not be deployed on the WebLogic Server on which the agent authentication provider has been installed. This is to ensure that such applications can independently enforce their own security requirements as necessary. The presence of agent authentication provider will interfere with the security evaluations done by such applications resulting in their malfunction.

Installing the Agent Filter in an Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following steps outline the process to install the agent filter for a given application:

1. If the application is currently deployed on the WebLogic Server, it must be removed using the WebLogic Server's Administration Console or WebLogic Server's deployment tools.
2. It is recommended that you create a backup of the deployment descriptor that will be edited in order to install the agent filter in this application.
3. Edit the application's `web.xml` deployment descriptor. Since filters were introduced in [Servlet Specification 2.3](#), the `web.xml`'s `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

4. Once the `<DOCTYPE>` element has been changed, add the `<filter>` elements in the deployment descriptor. This can be done by specifying the `<filter>` element and the `<filter-mapping>` element in the `web.xml` deployment descriptor immediately following the description element of the `<web-app>` element. The following is a sample `web.xml` with the `<filter>` and `<filter-mapping>` elements.

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>SunTM ONE Identity Server Policy Agent for WebLogic 7.0SP2</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

5. Once the `web.xml` deployment descriptor has been modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter has been added to the application.

Creating Role-to-Principal Mappings

Once the application has been configured to have the agent filter component in it, the agent filter will enforce authentication, thereby enabling the agent authentication provider to successfully resolve the role-to-principal mappings. However, these mappings must first be created in order that the hosted application may use them during runtime.

The following are two ways to create such mappings:

- Editing the WebLogic Server specific deployment descriptors.

The WebLogic Server specific deployment descriptors may be edited to create the role to principal mappings. These descriptors are in `weblogic.xml` and `weblogic-ejb-jar.xml` files. Refer WebLogic Server reference documentation to learn the details of how these descriptors may be edited to create the role to principal mappings. For sample descriptors, which create such mappings, refer to [Appendix B, “Sample Application Scenario” on page 345](#).

Using the WebLogic Server Administration Console.

The WebLogic Server administration console allows you to create the role to principal mappings by editing the deployed application's deployment descriptors on the fly. In order to use this facility, the application must be deployed and the WebLogic Server must be up and running at that time. Refer WebLogic Server documentation on how to use the Administration console to create such mappings for deployed applications.

Agent for PeopleSoft 8.3/8.4/8.8

Once you have installed the agent for PeopleSoft, you must perform a set of post-installation tasks. The following sections describe these post-installation tasks as well as list a few points that you must take note of before you start using the agent:

- [Configuring Sun ONE Web Server 6.0sp5 for Proxy Solution](#)
- [Installing the Agent Filter for the Deployed Application](#)
- [Configuring agentauthenticatorservlet](#)
- [Creating URL Policies in Sun ONE Identity Server to Allow Access to PeopleSoft Application](#)
- [Creating User Mapping in Sun ONE Identity Server](#)

- [Important Notes](#)

Configuring Sun ONE Web Server 6.0sp5 for Proxy Solution

The following procedure explains how to configure Sun ONE Web Server 6.0sp5 as proxy for BEA WebLogic Server that hosts PeopleSoft servlets. This procedure is required only if you selected the deployment option as proxy solution during the agent installation.

1. Install Sun ONE Web Server 6.0sp5 and the supported Sun ONE Identity Server policy agent. For more information on installing the policy agent for Sun ONE Web Server 6.0sp5, access the documentation at:

<http://docs.sun.com/db/doc/816-6772-10>

2. Download and install BEA WebLogic plug-in for Sun ONE Web Server. To do so, make a directory `libproxy_Dir`, for example `web_server_dir/server_instance/libproxy_dir`.

3. Go to `WebLogic_Dir/lib/platform`

4. Copy `libproxy.so` to `libproxy_dir` created in the above step. If you are installing the agent on HPUX 11, you must copy `libproxy.sl`. For detailed information, see:

<http://www.weblogic.com/docs51/admindocs/nsapi.html>

<http://e-docs.bea.com/wls/docs61/adminguide/nsapi.html>

5. Make the following modifications to the files `magnus.conf` and `obj.conf`, which are located at `web_server_dir/server_instance/config`

- o Append the following lines at the end of the file `magnus.conf`.

```
Init fn="load-modules" funcs="wl-proxy,wl-init"\
shlib=_libproxy_Dir/libproxy.so
Init fn="wl-init"
```

- Append the following lines at the end of the file `obj.conf`

```
<Object name="weblogic" ppath="*">
  Service fn=wl-proxy WebLogicHost=myserver.com\
  WebLogicPort=7001
</Object>
```

Object is given a name *weblogic*. This object instructs the web server to redirect all the URLs that match the pattern defined as the value of `ppath`. The library method invoked when such a pattern occurs, is defined as the value of the `fn` variable. The rest of the parameters are input parameters to the `wl-proxy` function itself.

The above example shows that all the requests to this server are serviced by the `wl-proxy` function. If you modify the `ppath=*/weblogic/*` then all URLs that matches the pattern `*/weblogic/*` are serviced by the `wl-proxy` function.

It is recommended that only the proxy web server be allowed to access the BEA WebLogic 5.1 server that hosts PeopleSoft servlets and all other accesses be blocked. You can do this using one of the following tasks:

- Installing a firewall on the machine executing the BEA WebLogic server with appropriate firewall rules. For example, you can allow `http` access only from the proxy web server machine, and deny all other `http` access.
- Enabling the `SimpleConnectionFilter` class on the BEA WebLogic server. For more information, refer the documentation at:

<http://www.weblogic.com/docs51/examples/security/net/package-examples.security.net.html>

Installing the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of tasks explains how the agent filter can be installed for the application you want to protect with the agent. This procedure is not required for PeopleSoft 8.3.

1. In order to install the agent filter, you must modify the deployed application's deployment descriptor. It is recommended that you create the necessary backups before modifying these descriptors.
2. Stop the BEA WebLogic Server instance that hosts PeopleSoft application using the following command:

```
$psftHome/bea-home/wlserver6.1/config/peoplesoft/stopPIA.sh
```

3. Edit the application's `web.xml` descriptor as follows:

```
[psftHome/bea-home/wlserver6.1/config/peoplesoft/applications/PORTAL/WEB-INF/web.xml]
```

Since filters were introduced in [Servlet Specification 2.3](#), `web.xml`'s `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as follows:

```
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
```

4. After modifying the `<DOCTYPE>` element, you must add the `<filter>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and the `<filter-mapping>` elements immediately following the description element of the `web-app` element in the descriptor `web.xml`. The following is a sample `web.xml` descriptor with the `<filter>` and the `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun™ ONE Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

5. Start the BEA WebLogic Server instance that hosts the PeopleSoft application using the following command:

```
$psftHome/bea-home/wlserver6.1/config/peoplesoft/startPIA.sh
```

Configuring agentauthenticatorservlet

Use the following steps to configure agentauthenticatorservlet. These steps are not required for PeopleSoft 8.3.

1. Edit the application's web.xml descriptor as follows to configure agentauthenticatorservlet:

```
[ <psftHome>/<bea-home>/wlserver6.1/config/peoplesoft/applications/PORTAL/WEB-INF/web.xml ]
```

2. Add the following code to web.xml:

```
<servlet>
  <servlet-name>agentauthenticatorservlet</servlet-name>
  <servlet-class>com.sun.identity.agents.peoplesoft.AgentAuthenticatorServlet</servlet-class>
  <load-on-startup>0</load-on-startup>
</servlet>
.....

<servlet-mapping>
  <servlet-name>agentauthenticatorservlet</servlet-name>
  <url-pattern>/agentauthenticatorservlet/*</url-pattern>
</servlet-mapping>
```

Creating URL Policies in Sun ONE Identity Server to Allow Access to PeopleSoft Application

The following policies must be created in Sun ONE Identity Server in order to allow access to PeopleSoft application:

Table 2-4 URL Policies to be created in Sun ONE Identity Server

PeopleSoft Application	Policies
HRMS 8.3	http://myhost.mydomain.com:8001/servlets/iclientservlet/* http://myhost.mydomain.com:8001/servlets/iclientservlet/piaSiteName/* http://myhost.mydomain.com:8001/servlets/cs/* http://myhost.mydomain.com:8001/servlets/agentauthenticatorservlet* http://myhost.mydomain.com:8001/*.gif

Table 2-4 URL Policies to be created in Sun ONE Identity Server

PeopleSoft Application	Policies
FIN 8.4 and HRMS 8.8	http://myhost.mydomain.com:8001/cs* http://myhost.mydomain.com:8001/PSAttachServlet* http://myhost.mydomain.com:8001/psc* http://myhost.mydomain.com:8001/SyncServer* http://myhost.mydomain.com:8001/psp* http://myhost.mydomain.com:8001/xmllink* http://myhost.mydomain.com:8001/SchedulerTransfer* http://myhost.mydomain.com:8001/psreports* http://myhost.mydomain.com:8001/agentauthenticatorservlet* http://myhost.mydomain.com:8001/*.gif

NOTE If you have custom servlets to be given access, similar policies must be created for them too.

Creating User Mapping in Sun ONE Identity Server

1. Log on to Sun ONE Identity Server Console as Administrator.
2. Select the LDAP attribute to be used to map Sun ONE Identity Server user ids to PeopleSoft user ids.
3. Update the attribute for each Sun ONE Identity Server user that needs access to PeopleSoft depending on the mode of operation you have chosen.

User Mapping Modes

The agent can be configured to work in any of the following modes using this configurable property in `AMAgent.properties`:

```
com.sun.am.policy.amPeopleSoft.user.mapping = USE_DN | HTTP_HEADER
| LDAP
```

- **USE_DN**

This is the default mode of operation. In this mode of operation, user Id attribute is used to determine the corresponding PeopleSoft user. In this case, user Id attribute in Sun ONE Identity Server should match the PeopleSoft user. Here, the following attribute becomes useless:

```
com.sun.am.policy.amPeopleSoft.user.attribute.name
```

- **HTTP_HEADER**

In this mode of operation, the PeopleSoft user mapping is obtained from a particular HTTP Header specified as the value of the following parameter:

```
com.sun.am.policy.amPeopleSoft.user.attribute.name = httpHeaderName
```

It is therefore required that either the filter or the agent installed on the proxying Sun ONE Web Server sets the name of the HTTP Header which contains the value of the PeopleSoft user. See “[Enabling LDAP Attributes](#)” on [page 174](#) for information on how to enable forwarding of LDAP Attributes as HTTP Headers. In case you are using Sun ONE Web Server as proxy, refer to documentation at following location:

<http://docs.sun.com/db/doc/816-6772-10>

NOTE This mode cannot be used in the case of HRMS 8.3, when the agent is working in the filter-mode (when the agent is deployed on PeopleSoft provided WebLogic Server).

- **LDAP**

In this mode of operation, PeopleSoft user mapping is obtained as the value of the LDAP attribute set for the following configurable parameter:

```
com.sun.am.policy.amPeopleSoft.user.attribute.name = employeename
```

In this case, the value of `employeename` LDAP Attribute will be used as the corresponding PeopleSoft user.

This mode of operation expects the correct `amAdmin` password in `serverconfig.xml`.

If you are using this mode of user mapping, the `amAdmin` password must be provided at the time of agent installation. In case you have not provided the correct `amAdmin` password, you can change it as follows:

1. Execute `agent_install_dir/SUNWamj2ee_agents/bin/agentadmin -encrypt plain text amAdmin password`

It will print the encrypted password as:

Encrypted Value => *encrypted_password*

2. Replace the `puser` and `dsameuser` passwords in the file `serverconfig.xml` with the *encrypted_password* value you just generated.

NOTE In a default installation of Sun ONE Identity Server, these two passwords are the same as the `amAdmin` password. If your Sun ONE Identity Server is customized and these passwords have been changed, use the tool `agentadmin` to generate an encrypted password for each of these users and replace the values with the same.

Important Notes

Read these points before you start using the PeopleSoft agent:

- After installing the agent, always start the application server domain using the following script installed by the agent:

```
psft_home/appserv/psadmin_domain_name
```

For example, `/usr/psft/appserv/psadmin_HDMO`

- PeopleSoft agent must always operate either in the `SSO_ONLY` or the `URL_POLICY` filter mode. Setting the filter mode to any other value may lead to the application being unreachable.
- Always invoke `agentadmin` utility as PeopleSoft user. For example `psft`
- To configure the agent to use some specific login module, do this:

- For PeopleSoft 8.3, set the property `com.sun.am.policy.amPeopleSoft.auth.module = Unix` and restart BEA WebLogic Server.
- For PeopleSoft 8.4/8.8, modify the following property too along with the above:

```
com.sun.am.policy.loginURL[0]
=https://myhost.mydomain.com:58080/amserver/UI/Login?module=Unix
```

- If you have enabled SSL on Sun ONE Identity Server and are using the JSSE supplied with the agent for SSL communications, then make sure that the following entry is added to the configuration file `psappsrv.cfg`:

```
JavaOptions =  
-Djava.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
```

The file `psappsrv.cfg` is located in the directory `$PS_HOME/appserv/DOMAIN/`.

- If the properties that are not hot-swap enabled are modified, you must restart both the BEA WebLogic Server and the PeopleSoft Application Server.

Verifying a Successful Installation

1. Start the application server domain using the following script installed by the agent:

```
psft_home/appserv/psadmin_domain_name
```

For example, `/usr/psft/appserv/psadmin_HDMO`

2. Start BEA WebLogic Server that hosts the PeopleSoft application.
3. Try to access the PeopleSoft application. You will be able to see the Sun ONE Identity Server login page.

On a successful login, Sun ONE Identity Server displays the PeopleSoft application with the appropriately mapped PeopleSoft user logged in. An unsuccessful login will not allow access to the PeopleSoft application.

Agent for Apache Tomcat Server 4.1.27

Once you have installed the agent for Apache Tomcat Server 4.1.27, you must install the agent filter to protect the web applications deployed on Tomcat Server. The agent filter can be deployed in any one of the following ways:

- [Installing the Filter in the Global Deployment Descriptor](#)
- [Installing the Filter in the Application's Deployment Descriptor](#)

Installing the Filter in the Global Deployment Descriptor

Unlike some other J2EE containers, Apache Tomcat Server provides you with an option to install a filter for all available resources and web applications at a single place in the server, using a global deployment descriptor (`web.xml`). If you had selected this option during installation— by default, this is selected— you can ignore this section as the necessary filter configuration code would have already been installed in the global deployment descriptor (`web.xml`) of your Tomcat server instance.

NOTE If you do select this option, all resources in the server will be protected by the filter and you will not be able to access even the welcome page of the server, if any, without providing the credentials and adding a rule to the Identity Server to access that resource. One way to handle this situation is to add your server's welcome page URL or any other resource that does not need the protection, to the `com.sun.am.policy.amFilter.notenforcedList[<index>]` property in the `AMAgent.properties` file. For details on this property, see [“Not-Enforced List” on page 172](#).

Installing the Filter in the Application's Deployment Descriptor

If you have chosen not to install the filter in the global deployment descriptor (`web.xml`), an agent filter must be installed in the web application's deployment descriptor (`web.xml`) to protect the application. It can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following sub-tasks explain how the agent filter can be installed for the application.

- To install the agent filter, you must ensure that the application is not currently deployed on the Tomcat Server. If it is currently deployed, you must remove it before proceeding any further.
- In order to install the agent filter, you will have to modify the deployed application's deployment descriptor (`web.xml`). It is therefore recommended that you create the necessary backups before proceeding to modify these descriptors.
- Edit the application's deployment descriptor (`web.xml`) as follows.

Ever since servlet filter was introduced in [Servlet Specification 2.3](#), the `<DOCTYPE>` element in a web application's deployment descriptor must be changed so that the deployment descriptor is compliant with Servlet 2.3 specification. This can be done by simply setting the `<DOCTYPE>` element as shown in the following figure:

```
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application2.3//EN"
"http://www.java.sun.com/dtd/web-app_2_3.dtd">
```

After modifying the `<DOCTYPE>` element, you now need to add the `<filter>` and `<filter-mapping>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and `<filter-mapping>` elements immediately before the `<listener>` or `<servlet>` element in `web.xml`. Please see the Web Application Deployment Descriptor DTD at http://www.java.sun.com/dtd/web-app_2_3.dtd for more information. The following is a sample `web.xml` with the `<filter>` and `<filter-mapping>` elements added.

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Agent for Oracle 9iAS R2 and Oracle 10g

Once you have installed the agent for Oracle 9iAS R2 or Oracle 10g, you must complete the following post-installation tasks:

- [Creating the Oracle Administrative Users and Roles in Sun ONE Identity Server](#)
- [Installing the Agent Filter for the Deployed Application](#)
- [Installing the Application Through Enterprise Manager Console](#)

The following sections provide a detailed description of these tasks.

Creating the Oracle Administrative Users and Roles in Sun ONE Identity Server

By default, the installation program configures a second agent to protect the Enterprise Manager. When used without the agent, the Enterprise Manager supports basic authentication and uses the default realm supplied with Oracle 9iAS R2 or Oracle 10g. The agent, however, modifies the Enterprise Manager's `web.xml` to support FORM-based authentication. In this scenario, when you try to authenticate with the Enterprise Manager for the first time, you will be redirected to Sun ONE Identity Server console to authenticate as Administrator. The following users and roles need to be present in Sun ONE Identity Server before any user tries to authenticate with Enterprise Manager and perform various operations supported through the Enterprise Manager console.

Table 2-5

User	Role	Comments
ias_admin	ias_admin	Needed to logon to Enterprise Manager.
admin	administrators	Needed to browse and perform various operations using the Enterprise Manager console.

For detailed steps on creating users and roles, and adding users to roles in Sun ONE Identity Server, refer Sun ONE Identity Server documentation at <http://docs.sun.com/db/prod/sl.slidsrv#hic>.

NOTE After creating the two users, do not forget to add the `ias_admin` user to the role `ias_admin` and the `admin` user to the role `administrators`.

Installing the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of tasks explains how the agent filter can be installed for the application you want to protect with the agent.

NOTE

- When installing the agent filter, you must ensure that the application is not currently deployed on the application server. If it is deployed, you must remove it before proceeding any further.
- In order to install the agent filter, you will have to modify the deployed application's deployment descriptor. It is therefore recommended that you create the necessary backups before proceeding to modify these descriptors.

1. Edit the application's web.xml descriptor as follows:

Since filters were introduced in [Servlet Specification 2.3](#), the web.xml's `<DOCTYPE>` element must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. This can be done by setting the `<DOCTYPE>` element as:

```

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

```

2. After modifying the `<DOCTYPE>` element, you now need to add the filter elements in the deployment descriptor. This can be done by specifying the `<filter>` and the `<filter-mapping>` elements immediately following the `<description>` element of the `<web-app>` element in the descriptor web.xml. The following is a sample web.xml descriptor with the `<filter>` and the `<filter-mapping>` elements added.

```

<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Identity Server Policy Agent</description>
    <filter-class> com.sun.identity.agents.filter.AmAgentFilter </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>

```

Once `web.xml` is modified to reflect the new `<DOCTYPE>` and filter elements, the agent filter is added to the application. You can now redeploy your application on the application server.

For more information on descriptors specific to Oracle 9iAS R2 and Oracle 10g, please refer to [“Assembly Descriptors” on page 350](#).

Installing the Application Through Enterprise Manager Console

Once the application is mapped to Sun ONE Identity Server roles and principals, the user can deploy the application through the Enterprise Manager console or the `dcmctl` tool. For steps, refer

http://download-west.oracle.com/docs/cd/A97329_03/web.902/a95880/config.htm#10472

07. As you progress through installation, you will notice that a custom User Manager (`com.sun.identity.agents.oracle.AmOracleUserManager`) has been installed for Enterprise Manager and is picked as a default option. This custom User Manager is the Identity Server-specific realm that is installed by the agent. If you choose to use any of the other two options/realms in this page, you will not be able to complete the installation.

Configuring the Agent for Enterprise Manager

While installing the agent for Oracle 9iAS R2 or Oracle 10g, the installation program configures a second agent in the `J2EE_POLICY` mode to protect Enterprise Manager (a web application for Oracle 9iAS R2 and Oracle 10g consoles). To learn more about the agent filter modes, see [“Agent Filter Modes” on page 156](#).

The installation program does not ask any questions about the protocol or the port of the Enterprise Manager. It assumes a default protocol (`http`) and a default port (1810) for Enterprise Manager. If, however, you want to change the default protocol and the port of Enterprise Manager, then you need to change two properties as shown below.

- The agent configuration file `AMAgent.properties`. This property is hot swappable; hence you do not have to restart Enterprise Manager for changes to take effect.

```
com.sun.am.policy.amFilter.port.check.map[1810]=http
```

For example, if Enterprise Manager is running on `https` instead of `http`, the property should be as follows:

```
com.sun.am.policy.amFilter.port.check.map[1810]=https
```

- The file `ORACLE_HOME/bin/emctl`. If you want session notifications for the modified port or protocol, you need to change the following `-D java` option in this file and restart `emctl`.

```
-Dcom.ipplanet.am.notification.url=http://hostname.domainname:1810/emd/n  
otification
```

For example, if Enterprise Manager is running on `https` instead of `http`, the `-D java` option should be set as follows:

```
-Dcom.ipplanet.am.notification.url=https://hostname.domainname:1810/emd/not  
ification
```

Agent for SAP Enterprise Portal 6.0 SP2

The following post-installation tasks must be performed before starting the SAP Enterprise Portal for the first time after the installation of the agent:

- [Removing Backed Up Files from the Server Directory](#)
- [Creating the Necessary URL Policies](#)
- [Configuring the User Mapping Mode](#)

Removing Backed Up Files from the Server Directory

When the agent is installed, it modifies various files in the SAP Enterprise Portal installation and takes backups where necessary. When the agent installation program makes multiple changes to a particular file, it may create more than one backup depending upon the installation requirements. These backup files can be easily identified by their suffix of the form `-preAgent-timestamp`. For example, if a file named `sample.xml` is modified by the installation program on `03/25/2004`, it may create a backup such as `sample.xml-preAgent-20040325`.

It is recommended that these sample files be removed from their respective directories and copied to another location outside the server directory before the SAP Enterprise Portal is started for the first time after the installation of the agent. This is necessary to ensure that none of these backup files is mistaken for a configuration file by the SAP Enterprise Portal runtime. Moreover, even though the files left behind are harmless, they can cause clutter in the long run and can be difficult to remove if the SAP Enterprise Portal has marked them for being stored in its database.

These files are located in the following directories. The *base-instance-dir* referred here indicates the location of *the* SAP EP6 instance.

- *base-instance-dir*
- *base-instance-dir/ume*
- *base-instance-dir/cluster/server*
- *base-instance-dir/cluster/server/managers*
- *base-instance-dir/cluster/server/services/servlet_jsp/work/jspTemp/irj/root/WEB-INF*

NOTE Do not delete these files. If the agent installation gets corrupted, these files are necessary for a manual restoration.

Creating the Necessary URL Policies

If the agent is installed and configured to operate in the URL Policy mode, the appropriate URL policies must be created. For instance, if the SAP Enterprise Portal is available on port 50000 using HTTP protocol, at least a policy must be created to allow access to the resource:

```
http://myhost.mydomain.com:50000/irj/*
```

If no policies are defined and the agent is configured to operate in the URL_POLICY mode, then no user will be allowed access to the SAP Enterprise Portal resources. Refer to the Identity Server Administration guide to learn how to create these policies using the Identity Server console or command line utilities.

Configuring the User Mapping Mode

After you complete the above steps, you must configure the user mapping mode used by the agent. This can be done by specifying the appropriate value for the property `com.sun.am.policy.amSAP.user.mapping` in the `AMAgent.properties` file. The following properties values can be set for this property:

```
com.sun.am.policy.amSAP.user.mapping = USE_DN
```

```
com.sun.am.policy.amSAP.user.mapping = LDAP
```

```
com.sun.am.policy.amSAP.user.mapping = HTTP_HEADER
```

When set to `USE_DN`, the agent uses the Identity Server user-id of the user to log him into the SAP Enterprise Portal. When set to `LDAP` or `HTTP_HEADER` mode, the SAP user-id is obtained by either reading a profile attribute for the user or by reading a header field in the request. The name of the profile attribute or the header field to be read by the agent can be configured by setting the value for the property

```
com.sun.am.policy.amSAP.user.attribute.name.
```

For more details on these and other properties, see “[AMAgent.properties Reference](#)” on page 179.

Agent for Macromedia JRun 4

After installing the agent for Macromedia JRun 4, perform the following post-installation tasks:

- [Installing the Agent Filter](#)
- [Starting and Stopping Macromedia JRun 4 Instance](#)

Installing the Agent Filter

The agent filter can be deployed in either of the following ways for the applications deployed on Macromedia JRun 4:

- [Installing the filter in the global deployment descriptor](#)
- [Installing the filter in the application’s deployment descriptor](#)

Installing the filter in the global deployment descriptor

Macromedia JRun Server 4.0 provides an option to install a servlet filter for all available resources and web applications at a single place in the server instance by using the global deployment descriptor `default-web.xml`. If you have selected to install the agent filter in `default-web.xml` during installation, ignore the following filter installation instructions as the installation program would have already installed the necessary filter configuration code in the global deployment descriptor (`default-web.xml`) of your server instance.

NOTE	If you select this option, all resources in the server instance will be protected by the servlet filter defined in <code>default-web.xml</code> and users will not be able to access even the welcome page of the server, if any, without providing the credentials and adding a rule to access that resource in Sun ONE Identity Server. One way to handle this problem is to add your server’s welcome page URL or any other resource that does not need the protection, to the property <code>com.sun.am.policy.amFilter.notenforcedList[<index>]</code> in the agent configuration file <code>AMAgent.properties</code> .
-------------	---

Installing the filter in the application's deployment descriptor

If you have chosen not to install the filter in the global deployment descriptor, an agent filter must be installed in the web application's deployment descriptor `web.xml` to protect the application. It can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following sub-tasks explain how the agent filter can be installed for the application:

- Before installing the agent filter, you must ensure that the application is not currently deployed on the Macromedia JRun Server instance. If it is already deployed, you must remove it before proceeding any further.
- In order to install the agent filter, you will have to modify the application's deployment descriptor `web.xml`. It is therefore recommended that you make the necessary backups before modifying this descriptor file.
- Edit the application's deployment descriptor `web.xml` as explained here.

Ever since servlet filter was introduced in [Servlet Specification 2.3](#), the `<DOCTYPE>` element in a web application's deployment descriptor is required to be changed to reflect that the deployment descriptor is compliant to Servlet 2.3 specification. This can be done by simply setting the `<DOCTYPE>` element as follows:

```
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application2.3//EN"
"http://www.java.sun.com/dtd/web-app_2_3.dtd">
```

After modifying the `<DOCTYPE>` element, you must add the `<filter>` and `<filter-mapping>` elements in the deployment descriptor. This can be done by specifying the `<filter>` and `<filter-mapping>` elements immediately before the first `<listener>` or `<servlet>` element in `web.xml`. See the [Web Application Deployment Descriptor DTD](#) for more information. The following is a sample `web.xml` with the `<filter>` and `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
```

```

</filter>
<filter-mapping>
  <filter-name>Agent</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
...
...
</web-app>

```

Starting and Stopping Macromedia JRun 4 Instance

After you have installed the policy agent for Macromedia JRun 4 Application Server instance, you will be unable to start, stop and restart the server instance using Macromedia JRun 4 Management Console. This is because, by default, Macromedia JRun 4 uses the same JVM configuration file `jvm.config` for all its instances. However, you can perform these tasks in the following ways:

- When you have multiple JRun instances, you would want to use the instance's own JVM configuration file to maintain their multiple JVM configurations. In such multiple JVM configurations, where there is a different JVM configuration for each instance, the JRun 4 server instance must be started from the command line using the `-config` option followed by the complete path name to the location of the JVM configuration file.

The policy agent installation program creates a new JVM configuration file called `jrun_instance_name_jvm.config` for each instance that is configured with this agent. This file is located in the directory `jrun_install_directory/bin`. You can use this file to start, stop or restart the JRun 4 server instance as explained in the following table. However, you can continue to use all other JRun4 command line options unchanged.

Table 2-6 Starting, Stopping and Restarting the Macromedia JRun 4 instances

Task	Command
Start	<code># jrun_install_directory/bin/jrun -config jrun_instance_name_jvm.config -start [instance_name]</code>
Stop	<code># jrun_install_directory/bin/jrun -config jrun_instance_name_jvm.config -stop [instance_name]</code>
Restart	<code># jrun_install_directory/bin/jrun -config jrun_instance_name_jvm.config -restart [instance_name]</code>

NOTE If no instance name is specified, each of the commands applies to all the configured instances.

- When you have only one Macromedia JRun 4 server instance, you can use the Macromedia JRun Management Console to start, stop and restart the instance by simply copying the *jrun_instance_name_jvm.config* to *jvm.config* file. However, if you uninstall or unconfigure the agent for this instance, you must make sure the original *jvm.config* file is restored after you have uninstalled or unconfigured the agent.

CAUTION When you uninstall the agent or unconfigure a Macromedia JRun 4 server instance, the *jrun_instance_name_jvm.config* file is simply removed. You must make sure to back up or save any custom configuration data that might have been added to this file.

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

Before starting the SAP Enterprise Portal or Web application Server for the first time after the agent has been installed, perform the following post-installation tasks.

- [Removing Backed up Files from the Server Directory](#)
- [Creating the Necessary URL Policies](#)
- [Configuring the User Mapping Mode](#)
- [Installing the Agent Filter](#)
- [Enabling Web-Tier Declarative Security](#)
- [Configuring the Library Reference for Application](#)

Removing Backed up Files from the Server Directory

When the agent is installed, it modifies various files in the SAP Web Application Server installation and Enterprise Portal installation and takes backups where necessary. When the agent installation program makes multiple changes to a particular file, it may create more than one backup depending upon the installation

requirements. These backup files can be easily identified by their suffix of the form `-preAgent-timestamp`. For example, if a file named `sample.xml` is modified by the installation program on `03/25/2004`, it may create a backup such as `sample.xml-preAgent-20040325`.

Remove these sample files from their respective directories and copy them to another location outside the server directory before the SAP Enterprise Portal or Web Application Server is started for the first time after the installation of the agent. This is necessary to ensure that none of these backup files is mistaken for a configuration file by the SAP runtime. Moreover, even though the files left behind are harmless, they can cause clutter in the long run and can be difficult to remove if the SAP runtime has marked them for being stored in its database.

These files are located in the following directories where *base-instance-dir* refers to the location of the SAP instance.

- *base-instance-dir*
- *base-instance-dir/ume* (only if Enterprise Portal is installed)
- *base-instance-dir/cluster/server*
- *base-instance-dir/cluster/server/managers*
- *base-instance-dir/cluster/server/services/servlet_jsp/work/jspTemp/irj/root/WEBINF* (only if Enterprise Portal is installed)
- *base-instance-dir/cluster/server/services/security*
- *base-instance-dir/cluster/server/services/security/work*

NOTE Do not delete these files. If the agent installation gets corrupted, these files are necessary for a manual restoration.

Creating the Necessary URL Policies

If the agent is installed and configured to operate in the URL Policy mode, the appropriate URL policies must be created. For instance, if the SAP Enterprise Portal is available on port 50000 using HTTP protocol, at least a policy must be created to allow access to the resource:

`http://myhost.mydomain.com:50000/*`

If no policies are defined and the agent is configured to operate in the `URL_POLICY` mode, then no user will be allowed access to the SAP Enterprise Portal resources or Web Application Server resources. Refer to the Identity Server Administration Guide to learn how to create these policies using the Identity Server console or command line utilities.

Configuring the User Mapping Mode

After you complete the above steps, you must configure the user mapping mode used by the agent. This can be done by specifying the appropriate value for the property `com.sun.am.policy.user.mapping` in the `AMAgent.properties` file. The following properties values can be set for this property:

```
com.sun.am.policy.user.mapping = USE_DN
```

```
com.sun.am.policy.user.mapping = LDAP
```

```
com.sun.am.policy.user.mapping = HTTP_HEADER
```

When set to `USE_DN`, the agent uses the Identity Server user ID of the user to log that user into SAP Enterprise Portal or Web Application Server. When set to `LDAP` or `HTTP_HEADER` mode, the SAP user ID is obtained by either reading a profile attribute for the user or by reading a header field in the request. The name of the profile attribute or the header field to be read by the agent can be configured by setting the value for the property `com.sun.am.policy.user.attribute.name`.

For more details on these and other properties, refer to [“AMAgent.properties Reference” on page 179](#) of this guide.

Installing the Agent Filter

This step is not needed if you intend to protect only the Enterprise Portal and do not have any other hosted applications on the underlying Web Application Server.

Before deploying any applications on the Web Application Server, you need to add the agent filter to the application's deployment descriptor `web.xml`. To add the agent filter:

1. If the application is deployed, remove it from the application server.
2. Create the necessary backups of the application's deployment descriptor since you need to modify these descriptors.

3. Change the web.xml <DOCTYPE>.

Since filters were introduced in Servlet Specification 2.3, the <DOCTYPE> element of web.xml must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. Reflect this compliance by setting the <DOCTYPE> element as follows:

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application2.3//EN"
    "http://www.java.sun.com/dtd/web-app_2_3.dtd">
```

4. Add the <filter> elements to the deployment descriptor.

Specify the <filter> and <filter-mapping> elements immediately following the description element of the <web-app> element in the descriptor web.xml. The following is a sample web.xml descriptor with the <filter> and the <filter-mapping> elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description> Identity Server Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the web.xml deployment descriptor is modified to reflect the new <DOCTYPE> and <filter> elements, the agent filter has been added to the application.

Enabling Web-Tier Declarative Security

This step is not needed if you intend to protect only the Enterprise Portal and do not have any other hosted applications on the underlying Web Application Server. In order for the agent to achieve single-sign on, enable web-tier declarative security for your protected applications. Enable this type of security by modifying the deployment descriptor `web.xml` and by modifying the agent configuration. Refer to [“Enabling Web-Tier Declarative Security” on page 163](#) for more information.

Configuring the Library Reference for Application

Once the Agent Filter has been added to the application you need to create a library reference in order to ensure that the filter classes are available for your application at runtime. Create the library reference by modifying the following file:

```
base-instance-dir/cluster/server/managers/reference.txt
```

1. Open the file in a text editor
2. Add the following line for every application the agent will protect:

```
reference <Application Name> library:AmSAPAgent
```

Where `<Application Name>` refers to the application that the agent will protect.

Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal

CAUTION The post-installation steps for this agent are unique. These steps are specifically tailored to BEA WebLogic 8.1 SP2/SP3 Portal. They are not tailored to BEA WebLogic 8.1 SP2/SP3 Server. Therefore, if you have installed the agent on WebLogic 8.1 SP2 Server or WebLogic 8.1 SP3 Server, do not follow these post-installation steps. Instead, follow the steps outlined in [“Agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1” on page 116](#).

After installing this agent on BEA WebLogic 8.1 SP2/SP3 Portal, attend to the following post-installation considerations and tasks:

- [Configure the Agent Authenticator Provider](#)
- [Application Configuration](#)
- [Install the Agent Filter in an Application](#)

- [Creating the WebLogic Users in Identity Server](#)
- [Creating the Necessary URL Policies](#)
- [Add the WebLogic Administrative User to the Realm Bypass List](#)
- [Customizing the WebLogic Portal Agent:](#)
- [Add an Entry to the Verification Handler Map Property \(Mandatory for WebLogic 8.1 SP3 Portal Applications Only\)](#)

Configure the Agent Authenticator Provider

The agent authentication provider can be added to the WebLogic Portal by using the WebLogic Server Console. This section outlines the steps necessary to add the agent authentication provider to the WebLogic Portal. The information provided in this section is only to facilitate the configuration of the agent authentication provider and is not a substitute for the information provided in WebLogic Portal documents. For a complete in-depth discussion on WebLogic Authenticators, refer WebLogic Portal documentation located at <http://www.bea.com>.

To configure the server to use the agent authentication provider:

1. Log on to the WebLogic Server console.
2. In the left frame, click
`agentdomain> Security> Realms> myrealm`

Where *agentdomain* is the domain you had configured. The myrealm selection is the default. It is not mandatory. Instead, you could configure a new realm and select it for configuration.
3. In the right frame, click the Providers tab.
4. Click Authentication.
5. Click Configure a New Agent Authenticator.
6. Click Create to create a new agent authenticator.
7. Change the control flag value from REQUIRED to OPTIONAL
8. Click Apply.
9. Go back to Agent Providers tab.
You should now be able to see an Agent Authenticator.
10. Click Default Authenticator.
11. Change the control flag from REQUIRED to OPTIONAL.

12. Click Apply.

NOTE Using the default, myrealm, to configure the agent is not mandatory. You could use a new realm to configure the agent. However, in the case of a new realm, make sure that the control flag value for the Agent Authenticator and any additional authentication providers are set to OPTIONAL.

13. Restart WebLogic Portal.

Application Configuration

The agent authentication provider component of this policy agent provides runtime mapping of various principals in Sun ONE Identity Server. Abstract security role names are used by the hosted application in order to determine if the currently authenticated user is authorized to access a particular resource or is otherwise a member of a given J2EE role. This runtime evaluation can occur only if the user is authenticated as a Sun ONE Identity Server principal by the means of Sun ONE Identity Server's authentication service. Without the user being authenticated appropriately, the results of such evaluations by the agent authentication provider will always be negative, resulting in access being denied to the user for the requested resource.

It is the agent filter component that enforces authentication for users who try to access particular application resources, thereby enabling the agent authentication provider to correctly evaluate the principal mappings as desired.

Unlike the agent authentication provider, which is installed in the core of WebLogic Portal, the agent filter is installed in the deployed application which must be protected by Sun ONE Identity Server. This is true for every portal application that must be protected on the WebLogic Portal using the agent. Applications that are not protected using the agent should not be deployed on the WebLogic Portal on which the agent authentication provider has been installed. This is to ensure that such applications can independently enforce their own security requirements as necessary. The presence of the agent authentication provider will interfere with the security evaluations by such applications, resulting in their malfunction.

Install the Agent Filter in an Application

Install the agent filter by modifying the deployment descriptor of the application that needs to be protected. To install the agent filter for a given portal application:

1. Edit the application's web.xml deployment descriptor.

Since filters were introduced in Servlet Specification 2.3, the `<DOCTYPE>` element of web.xml must be changed to reflect that the deployment descriptor is a Servlet 2.3 compliant deployment descriptor. Reflect this compliance by setting the `<DOCTYPE>` element as follows:

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application2.3//EN"
    "http://www.java.sun.com/dtd/web-app_2_3.dtd">
```

2. Add the `<filter>` elements to the deployment descriptor.

Specify the `<filter>` and `<filter-mapping>` elements immediately following the description element of the `<web-app>` element in the descriptor web.xml. The following is a sample web.xml descriptor with the `<filter>` and the `<filter-mapping>` elements added:

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Identity Server Policy Agent for WebLogic 8.1
Portal Application</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>Agent</filter-name>
    <url-pattern>*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the web.xml deployment descriptor is modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter has been added to the application.

Installing the agent filter should be done similar to other J2EE applications except that the `web.xml` for the portal application could have a set of pre-existing filters. Ensure that the agent filter element precedes all the other `<filter>` elements. Similarly, the filter mapping element should be before all the other `<filter-mapping>` elements. In practice, the agent filter should first intercept the request to properly enforce policies on the whole application.

Creating the WebLogic Users in Identity Server

Irrespective of the filter mode in which you run the agent, ensure that users who can log into the WebLogic Portal are also present in Identity Server. Users not present in either WebLogic Portal or Identity Server are denied access.

Creating the Necessary URL Policies

If the agent is installed and configured to operate in the `URL_POLICY/All` mode, the appropriate URL policies must be created. For instance, if the WebLogic Portal sample application is available on port 7001 using HTTP protocol, at least one policy must be created to allow access to the resource:

```
http://myhost.mydomain.com:7001/sampleportal/*
```

If no policies are defined and the agent is configured to operate in the `URL_POLICY/All` mode, then no user will be allowed access to the sample portal resources. Refer to the *Sun Java System Identity Server Administration Guide* to learn how to create these policies using the Identity Server Console or command-line utilities.

Add the WebLogic Administrative User to the Realm Bypass List

Add the WebLogic administrative user to the bypass principal list before starting the portal. The following is an example of how to add an administrative user named *weblogic* to the list:

```
com.sun.am.policy.amRealm.bypass.auth.principalList[0] = weblogic
```

This property will bypass authentication of the WebLogic 8.1 SP2/SP3 Portal administrative user with the agent authentication provider.

Customizing the WebLogic Portal Agent:

This section lists newly added properties, all of which can be customized. These properties are applicable to web applications running on WebLogic Server domains as well as WebLogic Portal domains. For the full list of J2EE policy agent properties and their descriptions, see [“AMAgent.properties Reference” on page 179](#).

The properties listed in this section are linked to their respective property descriptions in the “AMAgent.properties.Reference” section. These four properties can be changed in order to customize the portal agent:

- `com.sun.am.policy.user.mapping.mode`
- `com.sun.am.policy.user.attribute.name`
- `com.sun.am.policy.amFilter.logout.request.param.map`
- `com.sun.am.policy.amFilter.logout.uri.map`

Add an Entry to the Verification Handler Map Property (Mandatory for WebLogic 8.1 SP3 Portal Applications Only)

CAUTION This step is a required post-install step for WebLogic 8.1 SP3 Portal applications protected by the policy agent. Failure to perform this step can lead to malfunctioning of the agent, resulting in the application or portions thereof becoming inaccessible.

For each portal application running on WebLogic 8.1 SP3 portal domain, add an entry to the Verification Handler Map property as demonstrated in the following:

- The Verification Handler Map property:

```
com.sun.am.policy.amRealm.verification.handler.map
```

- An example of a portal web application is running on WebLogic 8.1 SP3 portal domain at the following URI:

```
/sampleportal
```

- An example of an entry that has been added to the Verification Handler Map property for the */sampleportal* application:

```
com.sun.am.policy.amRealm.verification.handler.map[sampleportal]=com.sun.agents.weblogic.AmWebLogicPortalVerificationHandler
```

For more information on this property see [“Verification Handler Map” on page 270](#).

Agent for Sun Java System Application Server

8.1

Once you have installed the agent for Sun Java System Application Server 8.1, complete the following post-installation tasks:

- [Install the Agent Filter for the Deployed Application](#)
- [Configure Domain Configuration Files for Remote Instances](#)

The following is a detailed description of these tasks:

Install the Agent Filter for the Deployed Application

The agent filter can be installed by simply modifying the deployment descriptor of the application that needs to be protected. The following list of sub-tasks explains how the agent filter can be installed for the application you want the agent to protect.

1. To install the agent filter, you must ensure that the application is not currently deployed on the application server. If it is currently deployed, you must remove it before proceeding any further.
2. In order to install the agent filter, you must modify the deployed application's deployment descriptor. Therefore, create the necessary backups before proceeding to modify these descriptors.
3. Edit the application's `web.xml` descriptor as follows:
 - a. Set the `<DOCTYPE>` element as shown in the following table.

```
<!DOCTYPE web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
```

Sun Java System Application Server supports the Java Servlet Specification version 2.4. Note that Servlet API version 2.4 is fully backward compatible with version 2.3, so all existing servlets should work without modification or recompilation. For more information refer to the *Sun Java System Application Server Developer's Guide*.

b. Edit the application's web.xml descriptor.

Add the `<filter>` elements in the deployment descriptor. Do this by specifying the `<filter>`, `<filter-mapping>`, and `<dispatcher>` elements immediately following the description element of the `<web-app>` element in the descriptor web.xml. The following table displays a sample web.xml descriptor with the `<filter>`, `<filter-mapping>`, and `<dispatcher>` elements added.

```
<web-app>
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Access Manager Policy Agent</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping id="FilterMapping_PolicyAgent">
    <filter-name>Agent</filter-name>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>ERROR</dispatcher>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

Once the web.xml deployment descriptor is modified to reflect the new `<DOCTYPE>` and `<filter>` elements, the agent filter is added to the application. You can now redeploy your application on Application Server.

NOTE The J2EE agent filter-class attribute is backward compatible. So, an application protected by a previous version of agent filter class (for example `com.sun.amagent.as.filter.AgentFilter`) does not need to modify web.xml to reflect the new filter class for the latest version of the agent.

Configure Domain Configuration Files for Remote Instances

For instances where Domain Administration Server (DAS) is not on the agent host, therefore DAS is remote, two files need to be changed: `server.policy` and `login.conf`.

To configure these files, manually add the lines displayed in the following code examples to the domain configuration files located on the host where DAS is running. Ensure that these lines do not already exist. If the configuration has been completed sometime in the past, the code might already exist.

Code Example 2-1 Configuration to server.policy File

```
ApplicationServer-base/domains/domain1/config/server.policy
grant codeBase "file: agent-base/SUNWam/j2ee_agents/
  lib/am_as81_agent_2_1.jar" {
  permission com.sun.appserv.security.ProgrammaticLoginPermission "login";
};
grant codeBase "file: agent-base/SUNWam/j2ee_agents/lib/am_agent_sdk_2_1.jar" {
  permission java.security.AllPermission;
};
grant codeBase "file: agent-base/SUNWam/j2ee_agents/lib/am_agent_filter_2_1.jar" {
  permission java.security.AllPermission;
};
grant codeBase "file: agent-base/SUNWam/j2ee_agents/lib/am_logging.jar" {
  permission java.security.AllPermission;
};
```

Code Example 2-2 Configuration to login.conf File

```
ApplicationServer-base/domains/domain1/config/login.conf
amAS81Realm {
  com.sun.identity.agents.as81.AmAS81LoginModule required;
};
```

Common Tasks

The following sections detail a few tasks that you need to perform to enable Cross Domain Single Sign On (CDSSO) after installing a J2EE agent. CDSSO is a feature that helps the J2EE agents to successfully achieve Single Sign-On in a multi-domain deployment scenario.

Enabling CDSSO

If the agent is installed in a domain different from the Sun ONE Identity Server, then CDSSO can be enabled by using the property called `com.sun.am.policy.amFilter.cdsso.enabled` available in the `AmAgent.properties` file. For detailed information on this property, see the section [“AMAgent.properties Reference” on page 179](#) in Chapter 3 of this guide.

Synchronizing the Agent Host and the Sun ONE Identity Server 6.1 Host

When using the J2EE agents, it is necessary to sync the agent host machine with the Sun ONE Identity Server host machine. This is important because of two main reasons:

- When the URL Policies are evaluated, the policy decisions have a time interval after which the decision expires.
- The AuthnResponse sent by the CDC Servlet in Sun ONE Identity Server 6.1 also sets a validity period for the assertion.

If the machines are not synced up, there is a great possibility that the requested resources are denied by the agent (even though URL policies allow access) because of the above mentioned reasons.

Moreover, when the J2EE agent is installed to work with Sun ONE Identity Server 6.1 and functions in the CDSSO mode, the AuthnResponse time validity is just 1 minute time interval. This may cause the agent to deny access because of expired assertions. To avoid this, increase the skew factor property for CDSSO `com.sun.am.policy.amFilter.cdsso.clock.skew` to a suitable value. This will expand the validity interval set for the assertion time. By default this value is 0.

For example, if you set the value for this skew factor property as 90 seconds, the validity interval of the assertion time will now be increased to 4 minutes (2 x 90 + 1 minute).

Customizing the Agent Installation

Now that you have the agent installation complete, you may want to customize it to suit your particular deployment scenario. The agent provides a rich set of properties that can be used to fine tune your deployment for performance as well as the desired degree of security for your deployed applications.

Certain features such as Web-Tier declarative security support require that you modify the agent configuration in order to enable them. To learn more on such features, please refer to [Chapter 3, “Agent Configuration” on page 153](#) to learn how the agent can be configured to suit your requirements.

Agent Configuration

This chapter explains how to enable and use various features of J2EE Policy Agents and also provides a complete description of the configuration parameters that are present in the agent configuration file `AMAgent.properties`.

The following topics are included in this chapter:

- [General Notes on the Agent Configuration File](#)
- [Agent Filter Modes](#)
- [Disabling the Agent Realm](#)
- [Hot-Swap Configuration](#)
- [Enabling Web-Tier Declarative Security](#)
- [Enabling Failover](#)
- [Login Attempt Limit](#)
- [Redirect Attempt Limit](#)
- [Not-Enforced List](#)
- [Enabling LDAP Attributes](#)
- [Configuring FQDN Handling](#)
- [Using Cookie Reset Functionality](#)
- [Enabling Port Check Functionality](#)
- [AMAgent.properties Reference](#)

Once the agent is installed on your system, you must perform certain agent configuration tasks to take advantage of a rich set of features that can customize the agent deployment for the benefit of the deployed and protected applications, the performance of the entire system, and for fine-tuning your security policies as per your security requirements. Some of the agent features such as Web-Tier declarative security support require you to modify this configuration in order to enable them.

The agent configuration is largely controlled by a set of properties in a file called `AMAgent.properties`. For Solaris platform installations, this file is located under a unique directory under `/etc/opt/SUNWam/j2ee_agents` directory based on the instance of application server that it applies to. On other platforms this file is located under a unique directory under `Agent_Install_Dir/j2ee_agents/config` directory.

General Notes on the Agent Configuration File

This section describes a few details that must be noted when referring to the agent configuration.

Hot-Swap Mechanism

Certain properties in this configuration file are hot-swap enabled. The value of these properties, when altered, are dynamically loaded by the agent such that it is not necessary to restart the application server for these changes to take effect. However, in cases where the property is explicitly identified as not enabled for hot-swap or in cases when the hot-swap mechanism is disabled on the system, the application server must be restarted for the changes to take effect.

List Constructs in the Configuration File

Certain properties in the `AMAgent.properties` file are specified as lists. A list construct is defined as follows:

Format:

`<key>[<index>]=<value>`

Where

`key` is the configuration key

`index` is a positive number starting from 0 that increments by 1 for every value specified in this list.

`value` is one of the values specified in this list.

NOTE

- Properties that are specified in this manner must follow the above format, otherwise they will be treated as invalid or missing properties.
 - More than one property can be specified for this key by changing the value of `<index>`. This value must start from the number 0 and increment by 1 for each entry added to this list.
 - If certain indices are missing, those indices are ignored and the rest of the specified values are loaded at adjusted list positions.
 - Duplicate index values result in only one value being loaded in the indexed or adjusted indexed position.
-

Example

```
com.sun.am.policy.example.list[0] = value0
com.sun.am.policy.example.list[1] = value1
com.sun.am.policy.example.list[2] = value2
```

Map Constructs in the Configuration File

Certain property keys in the `AMAgent.properties` file are specified as Maps. A Map construct is defined as follows:

Format

```
<key>[ <name>]=<value>
```

Where

`key` is the configuration key

`name` is a string that forms the lookup key as available in the Map

`value` is the value associated with the name in the Map

NOTE

- Properties that are specified in this manner must follow the above format, otherwise they will be treated as invalid or missing properties.
 - For a given <name> , there may only be one entry in the configuration for a given configuration key (<key>). If multiple entries with the same <name> for a given configuration key are present, only one of the values will be loaded in the system and the other values will be discarded.
-

Example

```
com.sun.am.policy.example.map[AL] = ALABAMA
com.sun.am.policy.example.map[AK] = ALASKA
com.sun.am.policy.example.map[AZ] = ARIZONA
```

Agent Filter Modes

The agent installation program as well as the agent configuration file allow you to set the agent filter in one of the five available modes of operation. Depending upon your security requirements, you may choose the mode that best suits your requirement. The following configuration property is used to control the mode of the agent filter:

```
com.sun.am.policy.amFilter.mode
```

The value for this property can be one among the following:

- NONE
- SSO_ONLY
- J2EE_POLICY
- URL_POLICY
- ALL

Regardless of what mode the agent filter is operating in, the agent realm will continue to function, if configured. This can therefore lead to a situation where the agent realm component may malfunction or may result in the negative evaluation of J2EE security policies configured in the application's deployment descriptors or being used through the J2EE programmatic security API. To avoid this, you may disable the agent realm component, if necessary. The sections that follow describe the different agent filter modes and also tell you how to disable the agent realm.

Agent Filter Mode - NONE

This mode of operation effectively disables the agent filter. When operating in this mode, the agent filter allows all requests to pass through. However, if the logging is enabled, the agent filter will still log all the requests that it intercepts.

NOTE This mode is provided to facilitate development and testing effort in a controlled development or test environment. It is highly recommended not to use this mode of operation in a production environment at any time.

Although this mode disables the agent filter from taking any action on the incoming requests other than logging, it has no effect on the agent realm that may still be configured in your application server and may get invoked by the deployed application if the deployed application has J2EE security policies in its descriptors or uses programmatic security. With the agent filter disabled, these applications will fail to evaluate the J2EE security policies correctly and as a result the deployed application may malfunction.

In order to fully disable the agent, you must therefore ensure that the agent realm is not active. Refer to the section [Disabling the Agent Realm](#) to find out how the agent realm can be disabled for your agent installation.

Once you have disabled the agent realm and the filter mode is set to NONE, it is functionally equivalent to not having the agent in your system at all.

NOTE When the agent filter is operating in this mode, any declarative J2EE security policy or programmatic J2EE security API calls will return a negative result regardless of the user.

Agent Filter Mode - SSO_ONLY

This is the least restrictive mode of operation for the agent filter. In this mode, the agent simply ensures that all users who try to access protected web resources are authenticated using Sun ONE Identity Server's authentication service. In this mode of operation the agent realm is not used and can be safely disabled. Refer to the section on [Disabling the Agent Realm](#) to find out how the agent realm can be disabled for your agent installation.

NOTE When operating in this mode, any declarative J2EE security policy or programmatic J2EE security API calls evaluated for the application will result in negative evaluation.

Agent Filter Mode - J2EE_POLICY

In this mode, the agent filter and agent realm together work with various Sun ONE Identity Server services to ensure the correct evaluation of J2EE Policies. These policies may be configured using the declarative security in the application's deployment descriptors, or may be implicit in the code of the application in the cases where it uses the J2EE programmatic security APIs. Any URL policies defined in Sun ONE Identity Server do not take effect in this mode of filter operation.

In case the deployed application uses declarative security in the web-tier, you must configure the agent to enable this feature. Refer to the section on [Enabling Web-Tier Declarative Security](#) to find out more details on how to enable this feature. When running in the J2EE_POLICY mode, the agent ensures that the security principal is set in the system for every authorized user access.

In the J2EE_POLICY mode, the agent will not enforce any applicable URL Policies as defined in Identity Server.

NOTE For IBM Websphere Application Server 5.0/5.1 agent, the web tier declarative security needs to be enabled, even if the applications use only programmatic security at the web tier. This is required for setting the principal at the web tier. If this is not enabled, the programmatic security APIs available through the `HttpServletRequest` interface will not work properly.

Agent Filter Mode - URL_POLICY

In this mode, the agent filter is used to enforce various URL policies that may be defined in Sun ONE Identity Server. This mode does not require the agent realm to be functional and can therefore be safely disabled. Refer to the section on [Disabling the Agent Realm](#) to find out how the agent realm can be disabled.

NOTE When the agent filter is in the URL_POLICY mode, the agent will not enforce any applicable J2EE declarative security policies. Such policies along with any calls to J2EE programmatic security APIs will return negative results.

Agent Filter Mode - ALL

This is the most restrictive mode of the agent filter. In this mode, the filter enforces both J2EE policies and URL policies as defined in Sun ONE Identity Server. This mode of operation requires that the agent realm be configured in the Application Server. When running in the ALL mode, the agent ensures that the security principal is set in the system for every authorized access.

It is strongly recommended that this mode of operation be used for deployed production systems.

Disabling the Agent Realm

In certain situations you may want to disable the agent realm to better suit your deployment requirements. The following sections explain how you can do this for your application server:

- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic Server 6.1 SP2](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1](#)
- [Agent for Oracle 9iAS and Oracle 10g](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

Use the following steps to disable the agent realm for Sun ONE Application Server 7.0:

1. Log in to the Sun ONE Application Server Administration Server.
2. Click on the node for the server name on which the agent has been installed. This expands the server menu on the left hand panel.
3. In the expanded tree under the server name node, click on the Security node. This displays the general security settings for this server instance on the right hand panel.

4. On the right hand panel, change the value of `Default Realm` to `file`. If the application server was configured to use a different realm than `file` before the agent was installed, you must choose that value from the pull-down menu instead.
5. Restart your Application Server to ensure the changes take effect.

The above mentioned steps disable the agent realm for Sun ONE Application Server. To enable the agent realm again, follow steps 1-3 and in step 4 change the value of `Default Realm` to `agentRealm`.

Agent for BEA WebLogic Server 6.1 SP2

Use the following steps to disable the agent realm for BEA WebLogic Server.

1. Log in to the WebLogic Administration Console.
2. On the left pane click the Security node. The console displays the Security configuration of the WebLogic Server in the right pane.
3. In the right pane, click the Filerealm tab. The console displays the details of the current File Realm.
4. In the form that is displayed on the right pane, under Caching Realm, select the value `none` from the pull down menu and click on the Apply button. If your server was configured to use a different caching realm before the agent was installed, you must choose that value from the pull down menu.
5. Restart the WebLogic Server for the changes to take effect.

The above mentioned steps disable the agent realm for BEA WebLogic Server. To enable the agent realm again, follow steps 1-3 and in step 4 change the value of Caching Realm to Agent Caching Realm.

Agent for IBM WebSphere Application Server 5.0/5.1

Use the following steps to disable the agent realm for WebSphere Application Server:

1. Log in to the WebSphere Administration Console.
2. Click on the Security node on the left hand panel. This expands the Security node on the left hand panel.

3. Under the Security node, click on the `Global Security` link. This displays the Global Security configuration screen on the right hand panel.
4. On the right hand panel either disable the checkbox `General Preferences -> Enabled`, or select a User Registry other than `Custom` in `General Preferences -> Active User Registry`.
5. Restart your Application Server to ensure that the changes take effect.

NOTE By disabling the checkbox in `General Preferences -> Enabled` on the Global Security Configuration page, you will be effectively disabling the security for the entire application server. It is recommended that you do this in a controlled development or test environment only and not in a production environment as it can lead to unauthorized access to system resources.

The above mentioned steps disable the agent realm for WebSphere Application Server. To enable the agent realm, follow steps 1-3 and in step 4, simply check the `Enabled` checkbox under `General Preferences -> Enabled` on the Global Security Configuration page and make sure that the `Active User Registry` is set to `Custom`.

Agent for Oracle 9iAS and Oracle 10g

Use the following steps to disable the agent realm for Oracle 9iAS and Oracle 10g:

1. Go to the OC4J instance directory and edit `application.xml`. The path to this file is typically as follows:
 - o For default OC4J Instance
`ORACLE_HOME/ora9ias/j2ee/home/config`
 - o For sample OC4J Instance
`ORACLE_HOME/ora9ias/j2ee/test/config`
2. Comment out the `<user-manager>` element in the XML. If you ever uninstall the agent, remember to uncomment the `<user-manager>` element before uninstalling the agent so that the agent can remove it using the uninstallation program.
3. Restart the OC4J instance through Enterprise Manager or any command line tool.

Agent for Sun Java System Application Server 8.1

Use the following steps to disable the agent realm for Sun Java System Application Server 8.1:

1. Log in to the Sun Java System Application Server 8.1 Administration console.
2. In the left-hand pane, select the Configurations node.
3. Select the *instance_name*-config node where your agent is installed.
where *instance_name* is the name of that particular server instance.
4. In the expanded tree under the server name node, click the Security node.
This displays the general security settings for this server instance on the right hand panel.
5. In the right-hand pane, change the value of `Default Realm` to `file`.
If the application server was configured to use a different realm than `file` before the agent was installed, you must choose that value from the pull-down menu instead.
6. Restart your Application Server to ensure the changes take effect.

The above mentioned steps disable the agent realm for Sun Java System Application Server 8.1. To enable the agent realm again, follow steps 1-4 and in step 5 change the value of `Default Realm` to `agentRealm`.

Hot-Swap Configuration

The agent supports hot-swap configuration that allows you to modify the agent configuration properties and have them take effect without requiring you to restart the application server on which the agent is installed. It should be noted that not all properties in the agent configuration file are hot-swap enabled, which implies that for properties that are not hot-swap enabled, you must restart the server to ensure that any changes to these property values take effect.

This mechanism is controlled by the following configuration property:

```
com.sun.am.policy.config.load.interval
```

The valid values for this property is any unsigned integer including 0, which indicates the amount of time in seconds after which the agent will check for changes to the configuration. This mechanism can be disabled by setting the value 0 for this configuration property.

This mechanism is primarily provided to facilitate the development and testing of your application in a controlled development or test environment. It is strongly recommended that this feature be disabled for production systems to ensure optimal utilization of system resources. Also, in a production system by disabling this feature, any accidental changes to the agent configuration will not take effect until the application server has been restarted.

The property that controls the hot-swap mechanism itself is hot-swap enabled. This means that if the hot-swap mechanism is enabled and you change the value of this property, the new value will take effect after the last hot-swap load interval expires. This can be therefore used to dynamically disable the entire hot-swap system. For example consider the following situation:

- The application server is started with the load interval set to 10 seconds. Therefore, in every 10 seconds any changes made to the agent configuration are picked up by the agent.
- Consider the case that you modify the load interval value while the application server is running and set it to 0. When the last load interval completes, the agent will pick up this new value. Since the value is set to 0 the agent will disable the hot-swap mechanism for the entire system.
- Once disabled, the configuration changes made to the agent configuration file will not be sensed by the agent. Therefore, even if you reset the value of this property now to any other number, it will not enable the hot-swap mechanism unless the application server is restarted.

In the case when the value of the load interval is set to 0 during the startup of the application server, the hot-swap mechanism will be disabled and cannot be enabled without restarting the server and ensuring that this value is set to a value greater than 0.

Enabling Web-Tier Declarative Security

Certain applications may require the use of web-tier declarative security that enforces role-based access control over web resources such as Servlets, JSPs, HTML files and any other resource that can be represented as a URI. This type of security is enforced by adding `security-constraint` elements to the deployed application's `web.xml` deployment descriptor.

Typically `security-constraint` elements are tied with `auth-constraint` elements that identify the role membership that will be enforced when a request for a protected resource is made by the client browser. The following example illustrates this idea:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Report Servlet</web-resource-name>
    <url-pattern>/ReportGenServlet</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>MANAGER</role-name>
  </auth-constraint>
</security-constraint>
```

This fragment of deployment descriptor can be used to ensure that access to the report generation servlet is allowed only to those users who are members of the role called `Manager`.

In order for such a construct to work, you must make the necessary modifications to the agent configuration file to ensure it can identify and handle such requests. This can be done by following these steps:

1. Ensure that a `login-config` element is specified for the web application that is being protected and that the `login-config` element has the `auth-method` set to `FORM`. The supporting `form-login-config` element is also required.
2. The `form-login-page` element of `form-login-config` should be added as one of the values for the following property in the agent configuration file:

```
com.sun.am.policy.amFilter.login.formList
```

As an example, consider the following `login-config` element of a protected application:

```
<login-config>
  <auth-method>FORM</auth-method>
  <form-login-config>
    <form-login-page>/jsp/login.jsp</form-login-page>
    <form-error-page>/block.html</form-error-page>
  </form-login-config>
</login-config>
```

Notice how the `form-login-page` is specified for the supporting `form-login-config` element. This value must be set for the following property in the agent configuration file as shown here:

```
com.sun.am.policy.amFilter.login.formList[0] =  
/Portal/jsp/login.jsp
```

Notice that the value of the `form-login-page` as specified in the deployment descriptor is not the same as what is specified in the agent configuration file. The difference being that when you enter this value in the agent configuration file, you must prefix it with the context path for the application on which this `form-login-page` is going to be used. In this particular example, the context path of the application is “/Portal”.

Similarly, if you have more than one application deployed that require web-tier declarative security, you must add their respective `form-login-pages` to the agent configuration file. For example, other entries could possibly be:

```
com.sun.am.policy.amFilter.login.formList[1] = /BankApp/SignOn  
com.sun.am.policy.amFilter.login.formList[2] = /ERP/LoginServlet
```

Please ensure that each such element added to this list has a unique index entry. Having duplicate indexed entries can result in the loss of data and consequently result in the malfunction of the application. To learn more about the way such list properties are specified in the agent configuration file, refer to the section on [General Notes on the Agent Configuration File](#).

Once you have configured the web application’s deployment descriptor to use the `form-login` mechanism for web-tier declarative security and have added the full URI of the `form-login-page` for each such application in the agent configuration file, the web-tier declarative security is enabled for these applications.

NOTE

- When a protected application is configured for web-tier declarative security handling by the agent, it must be redeployed with a form-login configuration as described in this section. This configuration requires that two application resources be specified in the application's `web.xml` deployment descriptor: one for the `form-login-page` and the other for the `form-error-page`. Regardless of whether the resource corresponding to the `form-login-page` exists in the application or not (this depends on how the agent is configured to handle the form-login requests), the resource corresponding to the `form-error-page` must be present in the application. This resource is directly invoked by the application server to indicate authentication failures and, optionally, authorization failures. If the application does not contain a valid `form-error-page` matching the URI specified in this deployment descriptor, it could result in HTTP 404 errors when the container chooses to display this error page.
 - A real `form-error-page` is not necessary for the applications protected by the policy agents for Oracle 9iAS R2 and Oracle 10g and for Tomcat Server 4.1.27, which support the property `com.sun.am.policy.amFilter.login.errorList`. This property can be used to configure the agent to handle `form-error-page` requests for applications that do not have a `form-error-page` to begin with.
 - For applications that do not contain a `form-login-page`, you can specify any URI as long as that URI does not conflict with any application resource and the matching value has been added to the configuration property `com.sun.am.policy.amFilter.login.formList`.
 - By default, the agent is configured to intercept all form-login requests and handle them without invoking the actual `form-login-page` resource as specified in the `web.xml` of the protected application. Thus, when using a default installation of the agent, the application is not required to have a resource corresponding to the `form-login-page` element specified in `web.xml`. This allows for the configuration of web-tier declarative security for applications that were not designed to use the form-login mechanism and instead relied on other login schemes available in J2EE specification. This behavior of the agent can be changed so that it allows the form-login requests to be handled by actual resources that exist within the application by changing the agent configuration properties as applicable. For details on how this can be done, please refer to the section [Customizing Agent Response for Form Login](#).
 - If the agent filter is operating in the URL_POLICY mode, any necessary URL policies to allow access to the `form-error-page` resource must be created for all users.
-

To further customize the behavior of the application when using web-tier declarative security, please refer to the details provided in the section [Web-Tier Security Details](#).

Configuring a Web Application to Enable Web-Tier Declarative Security

The instructions in this section are applicable only to the agents for Oracle 10g and Macromedia JRun 4.

All the applications that are protected by web-tier declarative security must define a form-login page that does a client-side redirect to itself. For example, if the form-login page `AMLogin.html` is deployed at the `<context-root>` for the web application `/SampleApp`, the page should contain the following `HEAD` element:

```
<HEAD>
  <meta http-equiv="refresh" content="0;url=/SampleApp/AMLogin.html">
  ...
</HEAD>
```

A sample form-login HTML page called `AMLogin.html` is shipped with the policy agent for reference and can be located in `Agent_Install_Dir/locale`. When using this sample file, make sure to substitute the `<CONTEXT_URI>` tag with the URI of the protected application and copy the file to the `<context-root>` of the web application.

```
<HEAD> <meta http-equiv="refresh" content="0;url=/SampleApp/AMLogin.html">
... </HEAD>.
```

Once you have made this change, you will have to include this file in the web application archive or make it available in the directory where the web application resides.

NOTE This sample file only serves as a reference for HTML-based form-login pages. If the form-login page format is different than HTML, the user must ensure to update their form-login pages.

Web-Tier Security Details

When the application server's web container gets a request for a resource that is protected by the web-tier declarative `security-constraint`, it must evaluate the credentials of the user against the agent realm to ensure that only authorized requests go through. In order to process such a request, the application server requires the user to sign on using the specified form login page as mentioned in the `form-login-config` element of the `web.xml` descriptor. Based on the specification of the `FORM` authentication mechanism, it is required that the user submits a valid user name as `j_username` and a valid password as `j_password` to the special URI `j_security_check` using the `HTTP POST` method of form submission.

The agent, once configured to support web-tier declarative security for the given application can isolate the request for accessing `form-login-page` and instead can stream out some data to the client browser. This data contains the user's login name and temporary encrypted password, which in turn uses Javascript to do automatic form submission as required. This gives the user a seamless Single Sign-On experience since the user does not have to re-login in order to access the protected resources for a deployed application that uses web-tier declarative security.

By default, the content that the agent sends to the client browser on intercepting a request for the form login page is read from the file called `FormLoginContent.txt` located in the `locale` directory of the agent installation. This file contains the following html code:

```
<html>
  <head>
    <title>Security Check</title>
  </head>
  <body onLoad="document.security_check_form.submit()">
    <form name="security_check_form" action="j_security_check" method="POST">
      <input type="hidden" value="am.filter.j_username" name="j_username">
      <input type="hidden" value="am.filter.j_password" name="j_password">
    </form>
  </body>
</html>
```

Before the agent streams out the contents of this file, it replaces all occurrences of the string `am.filter.j_username` by the appropriate user name. Similarly, all occurrences of the string `am.filter.j_password` are replaced by a temporary encrypted string that acts as a one-time password for the user.

Customizing Agent Response for Form Login

The agent configuration file allows you to completely control the contents that are sent out to the user when the application server requires a form login from the user.

Using the agent configuration file, you can customize the agent response in the following ways:

1. Modify the content of the `FormLoginContent.txt` file to suit your UI requirements as necessary. You must ensure that whatever modifications you do, the file must eventually submit the `j_username` and `j_password` to the action `j_security_check` via `HTTP POST` method.
2. You can specify the name of a different file using the property `com.sun.am.policy.amFilter.login.filename` in the agent configuration file. If you specify the file name, you must ensure that it exists within the `locale` directory of the agent installation.

If you wish that this file be used from another directory, you can simply specify the full path to this new file.

You must ensure that whatever modifications you do, the file must eventually submit the `j_username` and `j_password` to the action `j_security_check` via `HTTP POST` method.

3. If you have more than one application and would like to have an application-specific response to the form login requests, you can instruct the agent to allow the form login request to proceed to the actual form login page. This can be done by setting the value of the configuration property `com.sun.am.policy.amFilter.login.use.internal` as `false`.

In this situation, you must ensure that the resource that receives this request extracts the `am.filter.j_username` and `am.filter.j_password` from the `HttpServletRequest` as attributes and uses that to ensure that eventually a submit of these values as `j_username` and `j_password` is done to the action `j_security_check` via `HTTP POST` method.

The following JSP fragment demonstrates how this can be done:

```

<form action="j_security_check" method="POST">
<%
    String user = (String) request.getAttribute("am.filter.j_username");
    String password = (String) request.getAttribute("am.filter.j_password");
%>
    <ul>
        <li>Your username for login is: <b><%=user%></b></li>
        <li>Your password for login is: <b><%=password%></b></li>
    </ul>
    <input type="hidden" name="j_username" value="<%=user%>">
    <input type="hidden" name="j_password" value="<%=password%>">
    <input type="submit" name="submit" value="CONTINUE">
</form>

```

This mechanism would therefore allow you to have an application-specific form-login handling mechanism.

Enabling Failover

The agent allows basic failover capabilities. This helps you ensure that if the primary Sun ONE Identity Server for which the agent has been configured becomes unavailable, the agent will switch to the next Sun ONE Identity Server as specified in the agent configuration file.

This setup can be achieved by the following steps:

1. Provide a list of Sun ONE Identity Server Authentication services URLs that may be used by the agent to authenticate users who do not have sufficient credentials to access the protected resources. This can be done by using the following property:

```
com.sun.am.policy.amFilter.loginURL
```

You may specify more than one Login URL as follows:

```
com.sun.am.policy.amFilter.loginURL[0] = first_login_url
```

```
com.sun.am.policy.amFilter.loginURL[1] = second_login_url
```

```
com.sun.am.policy.amFilter.loginURL[2] = third_login_url
```

2. Provide a list of Sun ONE Identity Server Naming service URLs that may be used by the agent to get access to the various other service URLs that may be needed to serve the logged on user. This can be done by using the following property:

```
com.iplanet.am.naming.url
```

More than one naming service URL may be specified as a space delimited list of URLs. The following example illustrates this idea:

```
com.iplanet.am.naming.url = first_url second_url
```

NOTE	<p>The failover capability of the agent is limited to login and naming services. The URL Policy Service and Remote Logging service do not support failover in this release of the agent.</p> <p>This implies that the failover feature would work only when the agent filter mode is set to either <code>SSO_ONLY</code> or <code>J2EE_POLICY</code>. Certain other configuration settings may also be required in order to ensure that the failover mechanism works correctly. For more information, please refer to the Sun ONE Identity Server Programmer's Guide.</p>
-------------	---

Login Attempt Limit

When a user tries to access a protected resource without having authenticated with Sun ONE Identity Server's authentication services, the request is treated as a request with insufficient credentials. The default action taken by the agent when it encounters such a request is to redirect the user to the next available Login URL as configured in the agent configuration file.

It may be possible that despite the repeated redirects done by the agent, the user is unable to furnish the necessary credentials. In such a case, the agent can be directed to block such a request. This is configured using the Login Attempt Limit configuration property. The configuration property that controls this behavior is the following:

```
com.sun.am.policy.amFilter.login.attempt.limit
```

If a non-zero positive value is specified for this property in the agent configuration file, the agent will only allow that many attempts before it blocks the access request without the necessary credentials. When set to a value of zero, this feature is disabled.

It is strongly recommended to enable this feature in order to guard against potential denial-of-service attacks on your system.

Redirect Attempt Limit

The processing of requests by the agent can result in redirects for the client browser. Such redirects can happen when the user has not authenticated with Sun ONE Identity Server's authentication service, lacks the sufficient credentials necessary to access a protected resource and a variety of other reasons.

While the agent ensures that only the authenticated and authorized users get access to the protected resources, there is a remote possibility that due to misconfiguration of the system, the client browser may be put into an infinite redirection loop.

The Redirect Attempt Limit configuration property allows you to guard against such potential situations by ensuring that after a given number of consecutive requests from a particular user that result in the same exact redirect, the agent blocks the user request. This blocking of the request is only temporary and is removed the moment the user makes a request that does not result in the same redirect or results in access being granted to the protected resource. The configuration property that controls this feature is:

```
com.sun.am.policy.amFilter.redirect.attempt.limit
```

If a non-zero positive integer is specified as the value of this property, the agent will break the redirection loop after the specified number of requests result in the same redirects. When its value is set to zero, this feature is disabled.

It is highly recommended that this feature be enabled to protect the system from such situations. Further, it can also help in breaking potential denial of service attacks.

Not-Enforced List

The agent configuration file allows you to specify a list of URIs that are treated as not-enforced. Access to these resources is always granted by the agent. The configuration property that controls this list is as follows:

```
com.sun.am.policy.amFilter.notenforcedList
```

It is recommended that if your deployed application has pages that use a bulk of graphics that do not need the agent protection, such content be added to the agent's Not-Enforced list to ensure the optimal utilization of the system resources. Following is an example of the entries that you may specify in the Not-Enforced list:

```
com.sun.am.policy.amFilter.notenforcedList[0] = /images/*
```

```
com.sun.am.policy.amFilter.notenforcedList[1] = /public/*.html
com.sun.am.policy.amFilter.notenforcedList[2] = /registration/*
```

This will enable the agent to focus on enforcing access control only over requests that do not match these given URI patterns. The use of a wildcard (*) is allowed to indicate the presence of one or more characters in the URI pattern being specified.

Inverting the Not-Enforced List

In situations where only a small portion of the deployed application needs protection, you can configure the agent to do just that by inverting the Not-Enforced list. This results in the agent enforcing access control over the entries that are specified in the Not-Enforced list and allowing access to all other resources on the system. This feature is controlled by the following property:

```
com.sun.am.policy.amFilter.notenforcedList.invert
```

If you set the value `true` for this property, it would make the entries specified in the not-enforced list as enforced entries and the rest of the application resources will be treated as not-enforced.

CAUTION

When the not-enforced list is inverted, the number of resources for which the agent will not enforce access control are potentially very large. The use of this feature should therefore be done with extreme caution and only after extensive evaluation of the security requirements of the deployed applications.

NOTE

- If an Access Denied URI is specified, it will always be not-enforced by the agent regardless of the configuration of not-enforced list. This is necessary to ensure that the agent can use the Access Denied URI to block any unauthorized access for protected system resources.
 - When configuring access denied URIs within the deployment descriptor of the web application, you must ensure that these values are added to the Not-Enforced List of the agent. Failing to do so can result in application resources becoming inaccessible by the user.
 - Any resource that has been added to the Not-Enforced list must not access any protected resource. If it does so, it can result in unauthorized access to protected system resources. For example, if a servlet that has been added to the not-enforced list, in turn sends the request to another servlet, which is protected, it can potentially lead to unauthorized access to the protected servlet.
-

Enabling LDAP Attributes

Certain applications may rely on the presence of user-specific profile information in some form in order to process the user requests appropriately. The agent provides the functionality that can help such applications appropriately by making the LDAP attributes from the user's profile available in various forms.

The agent allows the fetching of LDAP Attributes and can make them available in one of the following three forms:

- HTTP Request Headers
- HTTP Request Attributes
- Cookies

The property that controls this feature is the following:

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

The value for this property can be one from `NONE`, `HEADER`, `REQUEST_ATTRIBUTE`, and `COOKIE`. Based on the set value, the agent retrieves the necessary LDAP attributes available for the logged on user and makes them available to the application appropriately. The attributes that are fetched by the agent and the names by which they are made available are controlled by the following property:

```
com.sun.am.policy.amFilter.ldapattribute.map
```

Using this property, you can specify any number of attributes that are required by the protected application. For example, if the application requires the attributes `cn` and `mail`, and it expects these attributes to be available under the names `COMMON_NAME` and `EMAIL_ADDR`, then your configuration setting would be as follows:

```
com.sun.am.policy.amFilter.ldapattribute.map[cn] = COMMON_NAME
com.sun.am.policy.amFilter.ldapattribute.map[mail] = EMAIL_ADDR
```

LDAP Attributes as HTTP Headers

When the agent is configured to provide the LDAP attributes as HTTP headers, these attributes can be retrieved using the following methods on the `javax.servlet.http.HttpServletRequest` interface:

```
long getDateHeader(java.lang.String name)
java.lang.String getHeader(java.lang.String name)
java.util.Enumeration getHeaderNames()
```



```
java.util.Enumeration getHeaders(java.lang.String name)
int getIntHeader(java.lang.String name)
```

The property that controls the parsing of a date value from an appropriate string as set in the LDAP attribute is the following:

```
com.sun.am.policy.amFilter.ldapattribute.date.format
```

This property defaults to the value `EEE, d MMM yyyy hh:mm:ss z` and should be changed as necessary.

Multi-valued attributes can be retrieved as an instance of `java.util.Enumeration` from the method:

```
java.util.Enumeration getHeaders(java.lang.String name)
```

LDAP Attributes as Request Attributes

When the agent is configured to provide the LDAP attributes as Request Attributes, the agent populates these attribute values into the `HttpServletRequest` as attributes that can later be used by the application as necessary. These attributes are populated as `java.util.Set` objects, which must be cast to this type before they can be successfully used.

LDAP Attributes as Cookies

When the agent is configured to provide the LDAP Attributes as cookies, the necessary values are set as server specific cookies by the agent with the path specified as `"/`.

Multi-valued attributes are set as a single cookie value in a manner that all values of the attribute are concatenated into a single string using a separator character that can be specified by the following configuration entry:

```
com.sun.am.policy.amFilter.ldapattribute.separator
```

It is the responsibility of the application to parse this value back into the individual values to ensure the correct interpretation of the multi-valued LDAP attributes for the logged on user.

If you are using the policy agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1, you may choose to encode the cookie values that are set using the value of the following property:

```
com.sun.am.policy.amFilter.ldapattribute.cookie.encode=true
```

This is required if the LDAP Attributes that you specify have special characters that are not allowed in cookie values. If the value of this property is set to `false` and if the LDAP Attributes contain special characters, the agent may behave inconsistently. Note that this property is applicable only to the agents for BEA WebLogic Server 7.0 SP2 and BEA WebLogic Server 8.1.

It is highly recommended that when you are using this mode of LDAP attributes, you should also be using the Cookie Reset functionality to ensure that these cookies get cleaned up from the client browser when the client browser's session expires. For more information, see the section [Using Cookie Reset Functionality](#).

Configuring FQDN Handling

To ensure appropriate user experience, it is necessary to enforce that the users use valid URLs to access resources protected by the agent. This functionality is controlled by two separate properties - the default FQDN specified as the property:

```
com.sun.am.policy.amFilter.fqdn.default
```

and the FQDN Map, which is specified as the following property:

```
com.sun.am.policy.amFilter.fqdn.map
```

The configuration property for default FQDN provides the necessary information needed by the agent to identify if the user is using a valid URL to access the protected resource. If the agent determines that the incoming request does not have a valid hostname in the URL, it redirects the user to the corresponding URL with a valid hostname. The difference between the redirect URL and the URL originally used by the user is only the hostname, which is now changed by the agent to a fully qualified domain name (FQDN) as per the value specified in this property.

The property FQDN Map provides another way by which the agent can resolve malformed access URLs used by the users and take corrective action. The agent gives precedence to entries defined in this property over the value defined in the default FQDN property. If none of the entries in this property matches the hostname specified in the user request, the agent uses the value specified for default FQDN property to take the necessary corrective action.

The FQDN Map property can be used for creating a mapping for more than one hostname. This can be done when the application server protected by this agent can be accessed using more than one hostname. As an example, consider a protected application server that can be accessed using the following hostnames:

- `www.externalhostname.com`
- `internalhostname.interndomain.com`

- *IP address*

In this case, assuming that `www.externalhostname.com` is the default FQDN, then the FQDN Map can be configured as follows to allow access to the application for users who will use the hostname `internalhostname.interndomain.com` or the raw IP address, say `192.101.98.45`:

```
com.sun.am.policy.amFilter.fqdn.map[internalhostname.interndomain.com] =
internalhostname.interndomain.com
```

```
com.sun.am.policy.amFilter.fqdn.map[192.101.98.45] = 192.101.98.45
```

Using Cookie Reset Functionality

The agent allows you to reset certain cookies that may be present in the user's browser session if the user's Sun ONE Identity Server session has expired. This feature is controlled by the following configuration properties:

```
com.sun.am.policy.amFilter.cookieReset.cookieList
```

```
com.sun.am.policy.amFilter.cookieReset.domainMap
```

```
com.sun.am.policy.amFilter.cookieReset.pathMap
```

The above three properties can be used to specify the exact details of the cookie that should be reset by the agent when a protected resource is accessed without a valid session.

The Cookie Reset List property specifies a list of cookie names that will be reset by the agent when necessary. Each entry in this list can correspond to a maximum of one entry in the Cookie Reset Domain Map property and the Cookie Reset Path Map property, both of which are used to define the cookie attributes - the domain on which a particular cookie should be set and the path on which it will be set.

When using this feature, you must ensure that the correct values of the domain and path are specified for every cookie entry in the cookie list. If these values are inappropriate, it may result in the cookie not being reset in the client browser.

When a cookie entry does not have an associated domain specified in the domain map, it is assumed that it is a server cookie. Similarly, when a cookie entry does not have a corresponding path entry specified, it is assumed that the cookie path is set to `/`.

Enabling Port Check Functionality

In situations when Sun ONE Identity Server and the application server are installed on the same system but on different ports, certain browsers may not send the HOST header correctly to the agent in situations where there are redirects involved between Sun ONE Identity Server's authentication service and the agent. In such situations, the agent, relying on the availability of the port number from the application server may misread the port number that the user is trying to access.

When such a situation occurs, it can have a severe impact on the system since the agent now senses a resource access that in reality did not occur and consequently the subsequent redirects as well as any policy evaluations may fail thereby making the protected application inaccessible to the end user.

This situation can be controlled by enabling Port Check functionality on the agent. This is controlled by the following configuration property:

```
com.sun.am.policy.amFilter.port.check.enable
```

When this property is set to `true`, the agent verifies the correctness of the port number read from the request against its configuration. The configuration that provides the reference for this checking is set by the following property:

```
com.sun.am.policy.amFilter.port.check.map
```

This property allows the agent to store a map of various ports and their corresponding protocols. When the agent is installed, this map is populated by the agent server's preferred port and protocol as specified during the installation. However, if the same agent is protecting more than one HTTP listeners, you must add that information to the map accordingly.

When the agent discovers an invalid port in the request, it takes corrective action by sending some HTML data to break the redirection chain so that the browser can reset its HOST header on the subsequent request. This content is read from the file that resides in the `locale` directory of agent installation. The name of the file is controlled by the following property:

```
com.sun.am.policy.amFilter.port.check.filename
```

This property can also be used to specify the complete path to the file that may be used to achieve this functionality. This file contains special HTML that uses a `META-EQUIV REFRESH` tag in order to allow the browser to continue automatically when the redirect chain is broken. Along with this HTML, this file must contain the string `am.filter.request.url`, which is dynamically replaced by the actual request URL by the agent.

You can modify the contents of this file or specify a different file to be used, if necessary, so long as it contains the above mentioned string that the agent can substitute in order to construct the true request URL with the correct port. The contents of this file should be such that it should either allow the user to automatically be sent to this corrected location or let the user click on a link or a button to achieve the same result.

AMAgent.properties Reference

The configuration file `AMAgent.properties` contains the necessary configuration properties needed for the agent to function correctly. It also contains the necessary information needed for the Sun ONE Identity Server SDK to function correctly in a client installation mode as used by the agent.

This section provides a listing of all agent configuration properties in the `AMAgent.properties` file with a brief explanation and examples.

NOTE Among the agent properties, the following are not available in version 2.1 of policy agents for BEA WebLogic 6.0SP2/7.0SP2/8.1, IBM WebSphere 5.0/5.1, and Sun ONE Application Server 7.0. For these agents, these properties were introduced in version 2.1.1.

- `com.sun.am.policy.amFilter.cdsso.enabled`
- `com.sun.am.policy.amFilter.cdsso.cdcServletURL[0]`
- `com.sun.am.policy.amFilter.cdsso.redirect.uri`
- `com.sun.am.policy.amFilter.cdsso.cookie.name`
- `com.sun.am.policy.amFilter.cdsso.clock.skew`
- `com.sun.am.policy.amFilter.login.referrer.map`

Also note that these properties are currently not available for PeopleSoft 8.3/8.4/8.8 agent.

- `com.sun.am.policy.amFilter.mode`
- `com.sun.am.policy.config.load.interval`
- `com.sun.am.policy.locale.language`
- `com.sun.am.policy.locale.country`
- `com.sun.am.policy.amAgent.moduleList`
- `com.sun.am.policy.amFilter.loginURL`

- `com.sun.am.policy.amFilter.counter.cookie.name`
- `com.sun.am.policy.amFilter.login.attempt.limit`
- `com.sun.am.policy.amFilter.ssotoken.urldecode`
- `com.sun.am.policy.amFilter.goto.parameter.name`
- `com.sun.am.policy.amFilter.session.binding`
- `com.sun.am.policy.amFilter.notenforcedList`
- `com.sun.am.policy.amFilter.notenforcedList.invert`
- `com.sun.am.policy.amFilter.notenforcedList.enableCache`
- `com.sun.am.policy.amFilter.notenforcedList.cacheSize`
- `com.sun.am.policy.amFilter.notenforcedList.cacheTime`
- `com.sun.am.policy.amFilter.accessDeniedURI`
- `com.sun.am.policy.amFilter.ldapattribute.date.format`
- `com.sun.am.policy.amFilter.ldapattribute.map`
- `com.sun.am.policy.amFilter.ldapattribute.mode`
- `com.sun.am.policy.amFilter.ldapattribute.separator`
- `com.sun.am.policy.amFilter.ldapattribute.cookie.encode`
- `com.sun.am.policy.amFilter.fqdn.default`
- `com.sun.am.policy.amFilter.fqdn.map`
- `com.sun.am.policy.amFilter.j2ee.auth.handler`
- `com.sun.am.policy.amFilter.j2ee.logout.handler`
- `com.sun.am.policy.amFilter.login.formList`
- `com.sun.am.policy.amFilter.login.use.internal`
- `com.sun.am.policy.amFilter.login.filename`
- `com.sun.am.policy.amFilter.login.preserve.referer`
- `com.sun.am.policy.amFilter.login.referer.map`
- `com.sun.am.policy.amFilter.cookieReset.enable`
- `com.sun.am.policy.amFilter.cookieReset.cookieList`
- `com.sun.am.policy.amFilter.cookieReset.domainMap`
- `com.sun.am.policy.amFilter.cookieReset.pathMap`
- `com.sun.am.policy.amFilter.audit.level`

- `com.sun.am.policy.amFilter.redirect.cookie.name`
- `com.sun.am.policy.amFilter.redirect.attempt.limit`
- `com.sun.am.policy.amFilter.legacy.support.enable`
- `com.sun.am.policy.amFilter.legacy.userAgent`
- `com.sun.am.policy.amFilter.legacy.redirect.uri`
- `com.sun.am.policy.amFilter.port.check.enable`
- `com.sun.am.policy.amFilter.port.check.map`
- `com.sun.am.policy.amFilter.port.check.filename`
- `com.sun.am.policy.amRealm.allow.fetch.all`
- `com.sun.am.policy.amRealm.peoplecontainer.level`
- `com.sun.am.policy.amRealm.organization.dn`
- `com.sun.am.policy.amAgentLog.disposition`
- `com.sun.am.policy.amAgentLog.local.file`
- `com.sun.am.policy.amAgentLog.local.file.rotate.enable`
- `com.sun.am.policy.amAgentLog.local.file.rotate.size`
- `com.sun.am.policy.amAgentLog.remote.file`
- `com.sun.am.policy.amRealm.bypass.auth.principalList`
- `com.sun.am.policy.amPeopleSoft.user.mapping`
- `com.sun.am.policy.amPeopleSoft.user.attribute.name`
- `com.sun.am.policy.amPeopleSoft.peoplecode.validate.sso`
- `com.sun.am.policy.amPeopleSoft.sstoken.urldecode`
- `com.sun.am.policy.amPeopleSoft.auth.module`
- `com.sun.am.policy.amPeopleSoft.goto.url`
- `com.sun.am.policy.amPeopleSoft.display.resource.root`
- `com.sun.am.policy.amPeopleSoft.display.resource.default`
- `com.sun.am.policy.amPeopleSoft.display.resource.map`
- `com.sun.am.policy.amFilter.cdsso.enabled`
- `com.sun.am.policy.amFilter.cdsso.cdcServletURL[0]`
- `com.sun.am.policy.amFilter.cdsso.redirect.uri`
- `com.sun.am.policy.amFilter.cdsso.cookie.name`

- `com.sun.am.policy.amFilter.cdsso.clock.skew`
- `com.sun.am.policy.amFilter.login.errorList`
- `com.sun.am.policy.amFilter.mode.map`
- `com.sun.am.policy.amRealm.filtered.roles.enable`
- `com.sun.am.policy.amSAP.user.mapping`
- `com.sun.am.policy.amSAP.user.attribute.name`
- `com.sun.am.policy.amSAP.lfe.cookie.name`
- `com.sun.am.policy.amSAP.goto.url`
- `com.sun.am.policy.amSAP.error.filename`
- `com.sun.am.policy.user.mapping`
- `com.sun.am.policy.user.mapping.mode`
- `com.sun.am.policy.user.attribute.name`
- `com.sun.am.policy.amFilter.logout.request.param.map`
- `com.sun.am.policy.amFilter.logout.uri.map`
- `com.sun.am.policy.amFilter.mapped.authentication.handler`
- `com.sun.am.policy.amRealm.verification.handler.map`
- `com.sun.am.policy.amRealm.verification.handler`
- `com.sun.am.policy.amFilter.logout.introspect.request.enable`
- `com.sun.am.policy.amFilter.logout.application.handler.map`

Agent Filter Mode

Property

`com.sun.am.policy.amFilter.mode`

Description

This property governs the mode of operation of the agent. The various modes of agent operation dictate the necessary processing that the agent would perform on a user request.

Valid Values

The valid value for this property can be one of the five strings: `NONE`, `SSO_ONLY`, `J2EE_POLICY`, `URL_POLICY`, and `ALL`. Any other value will be treated as an incorrect entry and the agent will default to the `ALL` mode of operation.

NOTE

Mode `NONE`

In this mode of operation, the agent does not take any action on the user request and is effectively disabled. Access to all user requests is therefore granted. Please read the Caution section after the description of mode `ALL`.

Mode `SSO_ONLY`

In this mode of operation, the agent enforces Single Sign On for the user using Sun ONE Identity Server Authentication Service.

Mode `J2EE_POLICY`

In this mode of operation, the agent enforces J2EE policies such as those configured in the deployment descriptors of the protected application or are implicit in the deployed application using the J2EE programmatic security API.

Mode `URL_POLICY`

In this mode of operation, the agent enforces URL policies as defined in Sun ONE Identity Server.

Mode `ALL`

In this mode of operation, the agent enforces the J2EE policies as well as the URL policies. Access to the protected resource is granted only if the user meets the success criteria for all applicable policies.

CAUTION

When this property is set to `NONE`, the agent will grant access to all protected resources. This mode of operation should not be used in deployed production systems at any time as it can result in unauthorized access to the protected system resources. This mode of operation is provided only to facilitate troubleshooting of the application in well controlled development and test environment and should not be used in any other environment.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

NA

Example

```
com.sun.am.policy.amFilter.mode = ALL
```

See Also

- [Agent Filter Modes](#)
- [J2EE Authentication Handler](#)

Configuration Reload Interval

Property

```
com.sun.am.policy.config.load.interval
```

Description

This property specifies the time in seconds after which the agent will poll the configuration file for changes. This configuration controls the entire hot-swap mechanism of the agent and may also be used to disable the hot-swap mechanism.

Valid Values

Unsigned integer value, including 0, which indicates the amount of time in seconds after which the agent will poll the configuration file for any changes.

NOTE

- When set to 0, the hot-swap mechanism will be disabled.
 - This property is hot-swap enabled, which means that the value of this property can be loaded dynamically. However, if the value is set to 0 at any time, the hot-swap mechanism will become disabled for the entire system and can only be enabled again by restarting the system.
 - During the system startup, if the value of this property is 0, the hot-swap mechanism will be disabled and cannot be enabled unless the system is restarted with this property set to a non-zero value.
 - It is recommended that the value for this property be set to 0 for deployed production systems to ensure optimal performance.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.config.load.interval = 300
```

See Also

[Hot-Swap Configuration](#)

Language Code

Property

```
com.sun.am.policy.locale.language
```

Description

This property specifies the Locale for agent operation.

Valid Values

Must be a valid ISO Language Code.

NOTE

For more information, refer to ISO 639 specification:

<http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>

Hot-Swap Enabled

No. If you change the value of this property, it will take effect only after you restart the application server.

Applicable Modes of Operation

NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.locale.language = en
```

See Also

[Country Code](#)

Country Code

Property

`com.sun.am.policy.locale.country`

Description

This property specifies the Locale for agent operation.

Valid Values

Must be a valid ISO Country Code.

NOTE

For more information, refer to ISO 3166 specification:

http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html

Hot-Swap Enabled

No. If you change the value of this property, it will take effect only after you restart the application server.

Applicable Modes of Operation

NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.locale.country = US
```

See Also

[Language Code](#)

Registered Module List

Property

`com.sun.am.policy.amAgent.moduleList`

Description

This property specifies the registered modules in the system. This property is set during the agent installation and should not be changed at any time.

Valid Values

As set during agent installation. Any other value is not valid.

NOTE

- This property is set during agent installation and must not be changed at any time.
 - This property must conform to the rules and format of LIST construct as described in ["List Constructs in the Configuration File"](#) on page 154 section.
-

Hot-Swap Enabled

No

Applicable Modes of Operation

`NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amAgent.moduleList[0] =  
com.sun.identity.agents.filter.AmFilterModule
```

See Also

[General Notes on the Agent Configuration File](#)

Login URL

Property

`com.sun.am.policy.amFilter.loginURL`

Description

This property specifies the login URLs to be used by the agent to redirect incoming users without sufficient credentials to Sun ONE Identity Server authentication service.

Valid Values

The complete URL to be used as the redirect URL in order to send users without sufficient credentials to Sun ONE Identity Server authentication service.

-
- | | |
|-------------|---|
| NOTE | <ul style="list-style-type: none">• This property is set during agent installation and need not be changed unless absolutely necessary.• This property must conform to the rules and format of LIST construct as described in the General Notes on the Agent Configuration File section. |
|-------------|---|
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.loginURL[0] =  
http://www.mycompany.com:58080/amserver/UI/Login
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Enabling Failover](#)

Counter Cookie Name

Property

```
com.sun.am.policy.amFilter.counter.cookie.name
```

Description

This property specifies the name of the cookie that will be used to track the number of unsuccessful login attempts made by the user.

Valid Values

The name of the cookie to be issued by agent in order to track the number of unsuccessful login attempts made by the user.

NOTE This property is set during agent installation and need not be changed unless absolutely necessary.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.counter.cookie.name = amFilterParam
```

See Also

- Discussion on [Login Attempt Limit](#)
- [Login Attempt Limit](#)

Login Attempt Limit

Property

```
com.sun.am.policy.amFilter.login.attempt.limit
```

Description

This property specifies the number of unsuccessful login attempts that a user can make using a single browser session, Once this limit is reached, the agent will block the user request.

Valid Values

Unsigned integer value, including 0, which indicates the number of unsuccessful login attempts that are allowed for any user trying to gain access to protected resources.

NOTE This option can be disabled by setting the value to "0"

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.login.attempt.limit = 5
```

See Also

- Discussion on [Login Attempt Limit](#)
- [Counter Cookie Name](#)

URL Decode SSO Token Flag

Property

```
com.sun.am.policy.amFilter.sstoken.urldecode
```

Description

This property indicates if the SSO Token needs to be URL decoded by the agent before it may be used.

Valid Values

- `true`: indicates that the token should be URL Decoded
- `false`: indicates that the token should not be URL Decoded

-
- NOTE**
- Valid but inappropriate value of this property may lead to the application being unreachable by the users.
 - The value of this property is set during the installation and must not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.ssotoken.urldecode = true
```

Goto Parameter Name

Property

```
com.sun.am.policy.amFilter.goto.parameter.name
```

Description

This property specifies the `goto Parameter name` to be used by the agent when redirecting the user to the appropriate authentication service. The value of this parameter is used by the authentication service to redirect the user to the original requested destination.

Valid Values

A string value that represents the `goto parameter name` as expected by the authentication service.

-
- NOTE**
- Invalid value of this property may result in the application becoming inaccessible.
 - The value of this property is set during the installation and must not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.goto.parameter.name = goto
```

Session Binding Flag

Property

```
com.sun.am.policy.amFilter.session.binding
```

Description

This property specifies if the agent will enforce session binding with authentication.

Valid Values

- `true`: indicates that session binding will be enforced
- `false`: indicates that session binding will not be enforced

NOTE

- When enforced, the agent will invalidate the user's local session in the application server if the user does not have a valid SSO token.
 - When enforced, the agent will create the user's local session in the application server if the user has a valid SSO token but does not have an active local session yet.
 - When not enforced, the agent will preserve the user's local session state in the application server irrespective of whether the user has an active local session or not.
 - The value of this property is set during the installation and must not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.session.binding = true
```

Not-Enforced List

Property

```
com.sun.am.policy.amFilter.notenforcedList
```

Description

This property specifies a list of patterns that can be used to evaluate if the requested URI does not require the protection enforced by the agent.

Valid Values

- Valid values must comply with the syntax of this property.
- The valid values can be exact URIs or patterns consisting of wild-card character '*' to indicate zero or more characters.

NOTE

- This property must conform to the rules and format of LIST construct as described in ["List Constructs in the Configuration File"](#) on page 154.
 - No default value is specified for this property during installation. Values for this property must be specified as necessary.
 - This property is commented out during agent installation and should be uncommented if its use is necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Examples

```
com.sun.am.policy.amFilter.notenforcedList[0]=*.gif
```

```
com.sun.am.policy.amFilter.notenforcedList[1]=/public/*
```

```
com.sun.am.policy.amFilter.notenforcedList[2]=/images/*
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Discussion on Not-Enforced List](#)
- [Not-Enforced List Inversion Flag](#)
- [Not-Enforced-List Cache Enable Flag](#)
- [Not-Enforced-List Cache Size](#)
- [Not-Enforced-List Cache Expiration Time](#)

Not-Enforced List Inversion Flag

Property

```
com.sun.am.policy.amFilter.notenforcedList.invert
```

Description

This property indicates if the not-enforced list will be interpreted as the enforced list by the agent, thereby treating all request URIs not present in this list as not-enforced.

Valid Values

- `true`: indicates that the not-enforced list will be interpreted as the enforced list by the agent.
- `false`: indicates that the not-enforced list will be interpreted as the not-enforced list by the agent.

NOTE

- Valid but inappropriate value of this property may lead to the application being unreachable by the users or may result in unauthorized access to protected system resources.
 - The value of this property should be changed only if necessary with extreme caution as it can possibly result in unauthorized access to the protected system resources.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.notenforcedList.invert = false
```

See Also

- Discussion on [Not-Enforced List](#)
- [Not-Enforced List](#)
- [Not-Enforced-List Cache Enable Flag](#)
- [Not-Enforced-List Cache Size](#)
- [Not-Enforced-List Cache Expiration Time](#)

Not-Enforced-List Cache Enable Flag

Property

```
com.sun.am.policy.amFilter.notenforcedList.enableCache
```

Description

This property specifies if the request URIs that are evaluated as enforced or not-enforced may be cached to increase performance of the system.

Valid Values

- `true`: indicates that caching is enabled
- `false`: indicates that caching is disabled

NOTE It is recommended that this property be set to `true` to increase the system performance.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.notenforcedList.enableCache = true
```

See Also

- Discussion on [Not-Enforced List](#)
- [Not-Enforced List](#)
- [Not-Enforced List Inversion Flag](#)
- [Not-Enforced-List Cache Size](#)
- [Not-Enforced-List Cache Expiration Time](#)

Not-Enforced-List Cache Size

Property

```
com.sun.am.policy.amFilter.notenforcedList.cacheSize
```

Description

This property specifies the number of entries that will be kept in the cache of not-enforced URIs and enforced URIs by the agent.

Valid Values

Non-zero unsigned integer indicating the number of enforced as well as not-enforced request URIs to be cached during runtime.

NOTE

- Values that are valid but not suited for the deployment scenario may result in degradation of system performance.
 - The value for optimal system performance will depend on the type of application deployed, the number of possible request URIs in the deployed application, the user load on the system, the expiration time set for cache entries and a host of other deployment specific factors.
 - To determine the most optimal value of this property, the application should be load tested in a controlled test environment before being deployed for production.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.notenforcedList.cacheSize = 1000
```

See Also

- [Discussion on Not-Enforced List](#)
- [Not-Enforced List](#)
- [Not-Enforced List Inversion Flag](#)
- [Not-Enforced-List Cache Enable Flag](#)
- [Not-Enforced-List Cache Expiration Time](#)

Not-Enforced-List Cache Expiration Time

Property

```
com.sun.am.policy.amFilter.notenforcedList.cacheTime
```

Description

This property specifies the amount of time in seconds that will be used to evaluate if a cached entry can be removed from the cache to free up resources for new cache entries.

Valid Values

Non-zero unsigned integer indicating the time in seconds that will be used as the cache expiration time for entries in the cache during cleanup operation.

NOTE

- Values that are valid but not suited for the deployment scenario may result in degradation of system performance.
 - The value for optimal system performance will depend on the type of application deployed, the number of possible request URIs in the deployed application, the user load on the system, the cache size set for cache entries and a host of other deployment specific factors.
 - To determine the most optimal value of this property, the application should be load tested in a controlled test environment before being deployed for production.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.notenforcedList.cacheTime = 300
```

See Also

- [Discussion on Not-Enforced List](#)
- [Not-Enforced List](#)
- [Not-Enforced List Inversion Flag](#)
- [Not-Enforced-List Cache Enable Flag](#)
- [Not-Enforced-List Cache Size](#)

Access Denied URI

Property

`com.sun.am.policy.amFilter.accessDeniedURI`

Description

This property specifies the Access Denied URI used by the agent to block unauthorized access requests.

Valid Values

The URI within the deployed application that must be used as the access denied URI to block incoming requests when necessary.

NOTE

- This property is common for all deployed applications.
 - This property must specify a URI that is always available on the system. Failure to do so can result in HTTP 404 errors being sent to the end-user.
 - In case when this property is not specified the agent uses HTTP Status Code 403 (Forbidden) to indicate a blocked access.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.accessDeniedURI = /errors/forbidden.html
```

See Also

Discussion on [Not-Enforced List](#)

LDAP Date Header Attribute Format String

Property

`com.sun.am.policy.amFilter.ldapattribute.date.format`

Description

This property specifies the format of Date/Time value to be expected as a result of an attribute lookup. This is required when using the specialized get methods of the `javax.servlet.http.HttpServletRequest` interface that return Date values for headers.

Valid Values

Valid `java.text.SimpleDateFormat` Time Format Syntax string. For more information, please refer to

<http://java.sun.com/j2se/1.3/docs/api/java/text/SimpleDateFormat.html>

NOTE

- The default value of this property is set to: `EEE, d MMM yyyy hh:mm:ss z` and should be changed as necessary.
 - Invalid value of this property may result in runtime exceptions in the application.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amFilter.ldapattribute.date.format = EEE, d MMM  
yyyy hh:mm:ss z
```

See Also

- Discussion on [Enabling LDAP Attributes](#)
- [LDAP Attribute Map](#)
- [LDAP Attribute Fetch Mode](#)

- [LDAP Attribute Cookie Separator Character](#)

LDAP Attribute Map

Property

`com.sun.am.policy.amFilter.ldapattribute.map`

Description

This property specifies the LDAP Attributes to be populated under specific names for the currently authenticated user.

Valid Values

- Valid values must comply with the syntax of this property.
- The specified LDAP Attribute should be a valid attribute.
- The specified name should be valid name for the specified mode of setting these attributes as mentioned in the configuration value for the property LDAP Attribute Fetch Mode.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - Please ensure that the specified Names do not conflict with existing names in the system for the specified fetch mode. Failing to do may result in the loss of the existing data in the system.
 - Any number of such properties may be specified as long as they are valid properties conforming to the above stated requirements.
 - This property is commented out during agent installation and should be uncommented if necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.ldapattribute.map[ cn ]=COMMON_NAME
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Discussion on Enabling LDAP Attributes](#)
- [LDAP Date Header Attribute Format String](#)
- [LDAP Attribute Fetch Mode](#)
- [LDAP Attribute Cookie Separator Character](#)

LDAP Attribute Fetch Mode

Property

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

Description

This property specifies how the LDAP Attributes are made available to the protected application.

Valid Values

The valid value for this property should be any one of the strings: NONE, HEADER, REQUEST_ATTRIBUTE, or COOKIE.

NOTE

- When set to `NONE`, the LDAP attributes are not fetched by the agent.
 - When set to `HEADER`, the LDAP attributes are populated as HTTP Headers. Multi-valued attributes are available in this mode using the method `javax.servlet.http.HttpServletRequest.getHeaders(java.lang.String name)`
 - When set to `REQUEST_ATTRIBUTE`, the LDAP attributes are populated as request attributes. These attributes are available as `java.util.Set` objects and can be obtained using the method `javax.servlet.http.HttpServletRequest.getAttribute(java.lang.String name)`
 - When set to `COOKIE`, the LDAP attributes are populated as cookies for the current user. Multi-valued attributes are available as a concatenated string separated by characters as defined in the property `LDAP Attribute Cookie Separator Character`.
 - When set to `COOKIE`, it is advisable that the corresponding cookie names be also added to the `Cookie Reset List` property and that the `Cookie Reset Enable` flag be switched to `true`.
 - Multi-valued attributes are available in all modes. Please refer to [Enabling LDAP Attributes](#) to find out more about how multi-valued attributes can be accessed from within a protected application.
 - This property value is set to `NONE` during agent installation and should be changed as necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amFilter.ldapattribute.mode = REQUEST_ATTRIBUTE
```

See Also

- Discussion on [Enabling LDAP Attributes](#)
- [LDAP Date Header Attribute Format String](#)
- [LDAP Attribute Map](#)
- [LDAP Attribute Cookie Separator Character](#)

- [Cookie Reset Enable Flag](#)
- [Cookie Reset List](#)
- [Cookie Reset Domain Map](#)
- [Cookie Reset Path Map](#)

LDAP Attribute Cookie Separator Character

Property

`com.sun.am.policy.amFilter.ldapattribute.separator`

Description

This property specifies a character, which is used by the agent to set the value of LDAP Attributes as cookies in the case where more than one value for the given attribute is available.

Valid Values

A single character that can be used by the agent to concatenate various values of a multi-valued LDAP attribute for the logged on user so that it can be set as a single cookie.

NOTE

- This property is not used when the value of the property LDAP Attribute Fetch Mode is set to any value other than `COOKIE`.
 - This property is not used for attributes that are single-valued.
 - The character specified by this property should not appear as a character within any LDAP attribute value.
 - This property is set during agent installation and should be changed as necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.ldapattribute.separator = |
```

See Also

- [Discussion on Enabling LDAP Attributes](#)
- [LDAP Date Header Attribute Format String](#)
- [LDAP Attribute Map](#)
- [LDAP Attribute Fetch Mode](#)
- [Cookie Reset Enable Flag](#)
- [Cookie Reset List](#)
- [Cookie Reset Domain Map](#)
- [Cookie Reset Path Map](#)

LDAP Attribute Cookie Encode

Property

```
com.sun.am.policy.amFilter.ldapattribute.cookie.encode
```

Description

This property specifies whether the LDAP Attribute needs to be encoded before being set as a cookie.

Valid Values

- `true`: indicates that attribute needs to be encoded before being set as a cookie.
- `false`: indicates that no encoding is done to the attribute before being set as a cookie.

NOTE

This property is not used when the value of the property LDAP Attribute Fetch Mode is set to any value other than COOKIE.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.ldapattribute.cookie.encode = true
```

See Also

- [LDAP Date Header Attribute Format String](#)
- [LDAP Attribute Map](#)
- [LDAP Attribute Fetch Mode](#)
- [LDAP Attribute Cookie Separator Character](#)
- [Cookie Reset Enable Flag](#)
- [Cookie Reset List](#)
- [Cookie Reset Domain Map](#)
- [Cookie Reset Path Map](#)

FQDN Default

Property

```
com.sun.am.policy.amFilter.fqdn.default
```

Description

This property specifies the default fully qualified name of the host where the agent is installed. This property is used by the agent to take corrective action in the case where the user may have typed in an incorrect URL such as by specifying partial hostname or using an IP address to access the protected resources.

Valid Values

The fully qualified domain name of the host where the agent is installed.

NOTE

- This property is required for the agent to function correctly. An invalid value for this property can lead to the system becoming inaccessible by the end users.
 - This property is set during the agent installation and should be changed only if absolutely necessary.
 - This property along with the values set in FQDN Map property govern how the agent handles access requests that do not present a fully qualified host name in the request URLs.
 - Mappings specified in the FQDN Map property take precedence over the value specified for this property.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.fqdn.default = www.mycompany.com
```

See Also

- Discussion on [Configuring FQDN Handling](#)
- [FQDN Map](#)

FQDN Map

Property

```
com.sun.am.policy.amFilter.fqdn.map
```

Description

The FQDN Map is a simple map that enables the agent to take corrective action in the case where the users may have typed in an incorrect URL such as by specifying partial hostname or using an IP address to access protected resources.

Valid Values

Valid values must comply with the syntax of this property which represent invalid FQDN values mapped to their corresponding valid counterparts.

-
- NOTE**
- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - The names used in this map are partial or incorrect FQDN Names that the user may use to access the system. The corresponding values are their valid counterparts.
 - Mappings specified in this property take precedence over the value specified for the Default FQDN property.
 - This property is commented out during agent installation and should be uncommented, if necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.fqdn.map[myserver]=myserver.mydomain.com
```

See Also

- [General Notes on the Agent Configuration File](#)
- Discussion on [Configuring FQDN Handling](#)
- [FQDN Default](#)

J2EE Authentication Handler

Property

```
com.sun.am.policy.amFilter.j2ee.auth.handler
```

Description

This property specifies the J2EE Authentication Handler to be used by the agent in order to authenticate the logged on user with the application server. This property is necessary for J2EE_POLICY mode and ALL modes of operation.

Valid Values

As set during agent installation. Any other value is not valid.

NOTE This property is set during agent installation and should not be changed at any time.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

See Also

- [Discussion on Agent Filter Modes](#)
- [Discussion on Enabling Web-Tier Declarative Security](#)
- [Agent Filter Mode](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)
- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

J2EE Logout Handler

Property

`com.sun.am.policy.amFilter.j2ee.logout.handler`

Description

This property specifies the J2EE Logout Handler to be used by the agent in order to log out the logged on user from the application server. This property is necessary for J2EE_POLICY mode and ALL modes of operation.

Valid Values

As set during agent installation. Any other value is not valid.

NOTE This property is set during agent installation and must not be changed at any time.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.j2ee.logout.handler =  
com.sun.identity.agents.filter.GenericJ2EELogoutHandler
```

See Also

- [J2EE Authentication Handler](#)
- [Agent Filter Mode](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)
- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

Login Form List

Property

```
com.sun.am.policy.amFilter.login.formList
```

Description

This property specifies the list of URIs that represent `form-login-page` as specified in the `form-login-config` elements of the protected application's deployment descriptors (`web.xml`) along with the context path of the deployed application. This property is used by the agent to intercept login request and take appropriate action.

Valid Values

- Valid values must comply with the syntax of this property.
- Valid values should be as specified in the `form-login-page` element of `web.xml` of the deployed application along with the context path of the application.

NOTE

- This property must conform to the rules and format of LIST construct as described in the [General Notes on the Agent Configuration File](#) section.
 - The value of `form-login-page` in the `web.xml` of the deployed application is only a part URI which must be preceded with the context path of the application. For instance, if the application context path is `/Portal` and the value of `form-login-page` is `/jsp/login.jsp`, then the actual value for this configuration becomes `/Portal/jsp/login.jsp`.
 - Failure to set this property correctly can result in malfunction of the agent resulting in the application or portions thereof becoming inaccessible.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Examples

```
com.sun.am.policy.amFilter.login.formList[0] =
/Portal/jsp/login.jsp
```

```
com.sun.am.policy.amFilter.login.formList[1] =
/BankApp/LoginForm.html
```

See Also

- [General Notes on the Agent Configuration File](#)

- [Discussion on Enabling Web-Tier Declarative Security](#)
- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [Form Login Use Internal Flag](#)
- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

Form Login Use Internal Flag

Property

`com.sun.am.policy.amFilter.login.use.internal`

Description

This property specifies if the agent must use internal content for handling form login requests.

Valid Values

- `true`: indicates that internal content will be used for form login requests.
- `false`: indicates that internal content will not be used for form login requests.

NOTE

- When set to `false`, the agent expects the application to handle form login requests in a manner that complies with the agent requirements. Please refer to the discussion on [Enabling Web-Tier Declarative Security](#) for more information on this mechanism.
 - When set to `true`, the agent uses internal content as specified in a file named `FormLoginContent.txt` available in the agent installation directory, which can be modified as required. Please refer to the discussion on [Enabling Web-Tier Declarative Security](#) for more information on this mechanism.
 - This property defaults to `true` and should only be changed if the corresponding form login pages have been modified as required by the agent. Please refer to the discussion on [Enabling Web-Tier Declarative Security](#) for more information on this mechanism.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.login.use.internal = true
```

See Also

- [General Notes on the Agent Configuration File](#)
- Discussion on [Enabling Web-Tier Declarative Security](#)
- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [Login Form List](#)
- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

Form Login Content File Name

Property

```
com.sun.am.policy.amFilter.login.filename
```

Description

This property specifies the name of the file that has the necessary content needed to handle form login requests.

Valid Values

- Name of a file that exists in the `locale` directory of the agent installation, or
- The complete path to a file that exists on the system and is readable by the server process on which the agent is installed.

NOTE

- This property does not apply if the value of the property Form Login Use Internal Flag is set to `false`.
 - The file specified by the value of this property must have read permissions for the server process on which the agent is installed.
 - The file specified by the value of this property must comply with the agent requirements needed to handle form login requests accordingly. Please refer to [Enabling Web-Tier Declarative Security](#) for further details on how this mechanism works.
 - This property is set during agent installation and should not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.login.filename =  
CustomFormLoginContent.txt
```

See Also

- Discussion on [Enabling Web-Tier Declarative Security](#)
- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)
- [Preserve Referer for Form Login Flag](#)

Preserve Referer for Form Login Flag

Property

```
com.sun.am.policy.amFilter.login.preserve.referer
```


Description

This property specifies if the referer should be used as the post authentication destination URL when a user does not have sufficient credentials to access the protected resource.

Valid Values

- `true`: indicates that the referer will be used as destination for post authentication redirect.
- `false`: indicates that the current resource will be used as destination for post authentication redirect.

NOTE

- This property is set during agent installation and should not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in the system becoming inaccessible or HTTP 404 errors being sent to the logged on user.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.login.preserve.referer = true
```

See Also

- [Discussion on Enabling Web-Tier Declarative Security](#)
- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)
- [Form Login Content File Name](#)

Default Referer Map for Form Login

Property

`com.sun.am.policy.amFilter.login.referer.map`

Description

This property specifies the application specific URI that will be used in case the referer is not set during the form-login sequence and the Preserve Referer flag has been enabled.

Valid Values

The URI to be used for a given application protected by this agent.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - The names used in this map are context path names of the application without the leading '/' character.
 - If the entry in this map applies to the default web application, the string “DEFAULT_WEB_APP” should be used for the name.
 - The values specified in this map can be any valid URIs on the protected application server.
 - This property is commented out during agent installation and should be uncommented if its use is necessary.
 - This property is not used if the value of the property Preserve Referer is set as false.
-

Hot-Swap Enabled

Yes

See Also

- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [J2EE Logout Handler](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)

- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

Examples

```
com.sun.am.policy.amFilter.login.referer.map[Portal]=/Portal/index.html
```

```
com.sun.am.policy.amFilter.login.referer.map[BankApp]=/BankApp/welcome.html
```

Applicable Modes of Operation

J2EE_POLICY, ALL

Cookie Reset Enable Flag

Property

```
com.sun.am.policy.amFilter.cookieReset.enable
```

Description

This property specifies if the Cookie Reset mechanism will be enabled for the agent. When enabled, the agent will reset various cookies as specified in by the property Cookie Reset List when the user tries to access a protected resource with insufficient credentials.

Valid Values

- `true`: indicates that the cookie reset mechanism is enabled.
- `false`: indicates that the cookie reset mechanism is disabled.

NOTE

- This property is set during agent installation and must not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in unauthorized access to protected system resources.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.cookieReset.enable = true
```

See Also

- Discussion on [Using Cookie Reset Functionality](#)
- [Cookie Reset List](#)
- [Cookie Reset Domain Map](#)
- [Cookie Reset Path Map](#)

Cookie Reset List

Property

```
com.sun.am.policy.amFilter.cookieReset.cookieList
```

Description

This property specifies a list of cookie names that must be reset when a user tries to access a protected resource with insufficient credentials. This property does not take effect if the [Cookie Reset Enable Flag](#) property is set to `false`.

Valid Values

Valid values must comply with the syntax of this property.

-
- | | |
|-------------|---|
| NOTE | <ul style="list-style-type: none">• This property must conform to the rules and format of LIST construct as described in "List Constructs in the Configuration File" on page 154.• This property is set during the agent installation and must not be changed unless absolutely necessary.• Invalid or inappropriate value for this property may result in unauthorized access to protected system resources. |
|-------------|---|
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.cookieReset.cookieList[0] = SESSIONID
```

See Also

- [General Notes on the Agent Configuration File](#)
- Discussion on [Using Cookie Reset Functionality](#)
- [Cookie Reset Enable Flag](#)
- [Cookie Reset Domain Map](#)
- [Cookie Reset Path Map](#)

Cookie Reset Domain Map

Property

```
com.sun.am.policy.amFilter.cookieReset.domainMap
```

Description

This property specifies a Map, which links the cookie names as specified in the Cookie Reset List property to domain names that would be used when resetting those cookies. This property does not take effect if the [Cookie Reset Enable Flag](#) property is set to *false*.

Valid Values

- Valid values must comply with the syntax of this property.
- Each value should be associated with the correct cookie name as specified in the property [Cookie Reset List](#). The domain names should conform to the requirements of HTTP Cookie domain names.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - This property is set during the agent installation and must not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in unauthorized access to protected system resources.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.cookieReset.domainMap[SESSIONID] =  
mycompany.com
```

See Also

- [General Notes on the Agent Configuration File](#)
- Discussion on [Using Cookie Reset Functionality](#)
- [Cookie Reset Enable Flag](#)
- [Cookie Reset List](#)
- [Cookie Reset Path Map](#)

Cookie Reset Path Map

Property

```
com.sun.am.policy.amFilter.cookieReset.pathMap
```

Description

This property specifies a Map, which links the cookie names as specified in the Cookie Reset List property to paths that would be used when resetting those cookies. This property does not take effect if the [Cookie Reset Enable Flag](#) property is set to `false`.

Valid Values

- Valid values must comply with the syntax of this property.
- Each value should be associated with the correct cookie name as specified in the property [Cookie Reset List](#). The path values should conform to the requirements of HTTP Cookie paths.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - This property is set during the agent installation and must not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in unauthorized access to protected system resources.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.cookieReset.pathMap[SESSIONID] = /
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Discussion on Using Cookie Reset Functionality](#)
- [Cookie Reset Enable Flag](#)
- [Cookie Reset Domain Map](#)

Audit Log Level

Property

```
com.sun.am.policy.amFilter.audit.level
```

Description

This property specifies the audit log level used by the agent that dictates the output of audit logs for various user accesses. This property applies to all accessed URLs in the system, which are protected by the agent.

Valid Values

The valid value for this property could be any one of the strings: `NONE`, `LOG_ALLOW`, `LOG_DENY`, or `LOG_BOTH`.

NOTE

- When set to `NONE`, no audit logs are recorded by the agent.
 - When set to `LOG_ALLOW`, only access requests that are granted are logged.
 - When set to `LOG_DENY`, only access requests that are denied are logged.
 - When set to `LOG_BOTH`, all access requests are logged.
 - Invalid value specified for this property would result in the agent defaulting to `LOG_BOTH`.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`NONE`, `SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.audit.level = LOG_ALLOW
```

See Also

- [Audit Log Disposition](#)
- [Audit Log Local File Name](#)
- [Audit Log Local File Rotate Flag](#)
- [Audit Log Local File Rotation Size](#)
- [Audit Log Remote File Name](#)

Redirect Counter Cookie Name

Property

```
com.sun.am.policy.amFilter.redirect.cookie.name
```

Description

This property specifies the name of the cookie that will be used to track the number of successive single point redirects made for the user.

Valid Values

A string that represents the name of the cookie to be issued by the agent in order to track the number of single point redirects made for the user.

NOTE	This property is set during agent installation and must not be changed unless absolutely necessary.
-------------	---

Hot-Swap Enabled

Yes

Applicable Modes of Operation

```
NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL
```

Example

```
com.sun.am.policy.amFilter.redirect.cookie.name = amFilterRDParam
```

See Also

- Discussion on [Redirect Attempt Limit](#)
- [Redirect Attempt Limit](#)

Redirect Attempt Limit

Property

```
com.sun.am.policy.amFilter.redirect.attempt.limit
```

Description

This property specifies the number of successive single point redirects that a user can make using a single browser session, which will trigger the blocking of the user request.

Valid Values

Unsigned integer value, including 0, which indicates the number of successive redirect attempts that are allowed for any user trying to gain access to protected resources.

-
- NOTE**
- This option can be disabled by setting the value to "0"
 - It is recommended to enable this mechanism by setting a non-zero value so that the system can be guarded against infinite redirection loops.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.redirect.attempt.limit = 10
```

See Also

- [Discussion on Redirect Attempt Limit](#)
- [Redirect Counter Cookie Name](#)

Legacy User Agent Support Flag

Property

```
com.sun.am.policy.amFilter.legacy.support.enable
```

Description

This property specifies if the agent protected applications will be accessed by legacy browsers. When enabled, the agent will be able to take specific action for such browsers to ensure that the features not supported by such browsers do not come in the way of agent functionality.

Valid Values

- `true`: indicates that legacy browser support is enabled.
- `false`: indicates that legacy browser support is disabled.

NOTE

- This property is set during agent installation and should not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in the system becoming inaccessible.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.legacy.support.enable = true
```

See Also

- [Legacy User Agent List](#)
- [Legacy User-Agent Intermediate Redirect URI](#)

Legacy User Agent List

Property

```
com.sun.am.policy.amFilter.legacy.userAgent
```

Description

This property specifies a list of User-Agent strings that may be used to identify legacy browsers. Certain legacy browsers do not handle the client side redirect request correctly when the redirect source and destination URLs are the same. In such cases, browsers identified by this list of User-Agent strings are sent to an intermediate location before being redirected to the original location.

Valid Values

- Valid values must comply with the syntax of this property.
- Each value in this list must be a valid pattern that can match the value of the HTTP header “user-agent” as sent by a legacy browser.
- The value string may also use a wild card character '*' to indicate one or more values together.

NOTE

- This property must conform to the rules and format of LIST construct as described in [“List Constructs in the Configuration File” on page 154](#).
 - This property is set during the agent installation and must not be changed unless absolutely necessary.
 - Invalid or inappropriate value for this property may result in excessive redirects for the logged on user, which may degrade the overall system performance.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.legacy.userAgent[0] = Mozilla/4.7*
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Legacy User Agent Support Flag](#)
- [Legacy User-Agent Intermediate Redirect URI](#)

Legacy User-Agent Intermediate Redirect URI

Property

`com.sun.am.policy.amFilter.legacy.redirect.uri`

Description

This property specifies the URI to which the agent will redirect the legacy browsers if necessary.

Valid Values

A valid URI that belongs to the application, which is the primary application for this server.

NOTE

- Invalid value specified for this property can result in the application becoming inaccessible.
 - This property does not come into effect if there are no values specified for the property Legacy User-Agent List or the value specified for the property Legacy User Agent Support flag is `false`.
 - This property is set during agent installation and must not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.legacy.redirect.uri = /Portal/dummy_uri
```

See Also

- [Legacy User Agent Support Flag](#)
- [Legacy User Agent List](#)

Port Check Enable Flag

Property

`com.sun.am.policy.amFilter.port.check.enable`

Description

This property specifies if the agent will use the default port mapping as specified in the property Port Map to resolve issues arising from incorrect HOST header values as sent by various browsers.

Valid Values

- `true`: indicates that the port mapping is enabled.
- `false`: indicates that the port mapping is disabled.

NOTE

- Invalid or inappropriate value specified for this property can result in the system becoming inaccessible to certain browsers.
 - This property is set during agent installation and must be changed only if absolutely necessary.
 - This property is useful when the agent and the Sun ONE Identity Server authentication service are installed on the same host but different ports. In such situations, certain browsers may not reset the HOST header correctly when being redirected back to the agent-protected resource.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.port.check.enable = true
```

See Also

- Discussion on [Enabling Port Check Functionality](#)
- [Port Check Map](#)

- [Port Check Content File Name](#)

Port Check Map

Property

`com.sun.am.policy.amFilter.port.check.map`

Description

This property specifies a Map of various ports where the agent-enabled listeners operate. This property is necessary to ensure that the resource policies are evaluated correctly even if the client provided `HOST` header value is invalid.

Valid Values

- Valid values must comply with the syntax of this property.
- The names specified for this map should be valid port numbers and the corresponding values must either be `http` or `https`.

NOTE

- This property must conform to the rules and format of MAP construct as described in ["Map Constructs in the Configuration File" on page 155](#).
 - Please ensure that the listeners operating on each of these ports match the protocol scheme that is mentioned in their values.
 - Invalid or inappropriate value specified for this property can result in the system becoming inaccessible.
 - This property is set during agent installation and should be changed as necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amFilter.port.check.map[80] = http
```

See Also

- [General Notes on the Agent Configuration File](#)
- [Enabling Port Check Functionality](#)
- [Port Check Enable Flag](#)
- [Port Check Content File Name](#)

Port Check Content File Name

Property

`com.sun.am.policy.amFilter.port.check.filename`

Description

This property specifies the name of the file that has the necessary content needed to handle requests that need port correction.

Valid Values

- Name of a file that exists in the `locale` directory of the agent installation, or
- The complete path to a file that exists on the system and is readable by the server process on which the agent is installed.

NOTE

- This property does not apply if the value of the property Port Check Enable Flag is set to false.
 - The file being pointed to by the value of this property must have read permissions for the server process on which the agent is installed.
 - The file being pointed to by the value of this property must comply with the agent requirements needed to handle invalid port requests accordingly. Please refer to the section [Enabling Port Check Functionality](#) for further details of how this mechanism works.
 - This property is set during agent installation and should not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.port.check.filename =  
CustomPortCheckContent.txt
```

See Also

- Discussion on [Enabling Port Check Functionality](#)
- [Port Check Enable Flag](#)
- [Port Check Map](#)

Fetch All Operation Flag

Property

```
com.sun.am.policy.amRealm.allow.fetch.all
```

Description

This property specifies if the agent should allow a fetch all operation, which may be needed to achieve console integration of the application server with Sun ONE Identity Server.

Valid Values

- `true`: indicates that fetch all operation is allowed
- `false`: indicates that fetch all operation is not allowed

NOTE

- When set, the list of users and roles as defined in Sun ONE Identity Server may become available in the application server's console.
 - Since this setting can enable the application server administrator to see Sun ONE Identity Server principals in the console, it should be used with caution.
 - The default value of this setting is set during agent installation and should be changed as necessary.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amRealm.allow.fetch.all = false
```

People Container Level

Property

```
com.sun.am.policy.amRealm.peoplecontainer.level
```

Description

This property specifies the people container level to be used when searching for principals in Sun ONE Identity Server.

Valid Values

Non-zero unsigned integer representing the People Container Level in Sun ONE Identity Server, which may be used when searching for principals.

NOTE This property is set during agent installation and need not be changed unless absolutely necessary.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amRealm.peoplecontainer.level = 1
```

See Also

[Organization DN](#)

Organization DN

Property

`com.sun.am.policy.amRealm.organization.dn`

Description

This property specifies the DN for the organization, which will be used to search principals in Sun ONE Identity Server.

Valid Values

String representing the organization DN for the organization for which this agent has been installed.

NOTE

This property is set during agent installation and must not be changed unless absolutely necessary.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`SSO_ONLY`, `URL_POLICY`, `J2EE_POLICY`, `ALL`

Example

```
com.sun.am.policy.amRealm.organization.dn = dc=example,dc=com
```

See Also

[People Container Level](#)

Audit Log Disposition

Property

`com.sun.am.policy.amAgentLog.disposition`

Description

This property specifies the disposition of audit log that the agent will use when writing audit log messages.

Valid Values

The valid values for this property is any string from the following: LOCAL, REMOTE, ALL

NOTE

- When set to LOCAL, the audit logs are kept on the local system where the agent is installed.
 - When set to REMOTE, the audit logs are sent to the remote system where the Sun ONE Identity Server Log Service is available.
 - When set to ALL, the audit logs are kept on the local system as well as sent to the remote system where the Sun ONE Identity Server Log Service is available.
 - In the event that the value is set to REMOTE and the agent is unable to send the log message to the remote system, the log message will be written to the local system where agent is installed.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amAgentLog.disposition = REMOTE
```

See Also

- [Audit Log Level](#)
- [Audit Log Local File Name](#)

- [Audit Log Local File Rotate Flag](#)
- [Audit Log Local File Rotation Size](#)
- [Audit Log Remote File Name](#)

Audit Log Local File Name

Property

`com.sun.am.policy.amAgentLog.local.file`

Description

This property specifies the file name used by the agent to capture audit logs on the system where the agent is installed.

Valid Values

String representing the complete path name of the file to be used by agent to record audit messages.

-
- | | |
|-------------|--|
| NOTE | <ul style="list-style-type: none">• Please ensure that the application server process has sufficient permissions to be able to write to this file.• Invalid value specified for this property can result in the loss of audit messages or can result in the system becoming inaccessible. |
|-------------|--|
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amAgentLog.local.file = /var/opt/logs/agent.log
```

See Also

- [Audit Log Level](#)

- [Audit Log Disposition](#)
- [Audit Log Local File Rotate Flag](#)
- [Audit Log Local File Rotation Size](#)
- [Audit Log Remote File Name](#)

Audit Log Local File Rotate Flag

Property

`com.sun.am.policy.amAgentLog.local.file.rotate.enable`

Description

This property specifies if the file used by the agent for capturing audit messages on the local system should be rotated or not.

Valid Values

- `true`: indicates that the file should be rotated
- `false`: indicates that no rotation is necessary

NOTE Default value of this property is `false` and should be changed as necessary.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amAgentLog.local.file.rotate.enable = true
```

See Also

- [Audit Log Level](#)
- [Audit Log Disposition](#)

- [Audit Log Local File Name](#)
- [Audit Log Local File Rotation Size](#)
- [Audit Log Remote File Name](#)

Audit Log Local File Rotation Size

Property

`com.sun.am.policy.amAgentLog.local.file.rotate.size`

Description

This property specifies the size (in bytes) of the file used by the agent for capturing audit messages on the local system, beyond which this file should be rotated. This property does not take effect unless the value of the property [Audit Log Local File Rotation Flag](#) is set to `true`.

Valid Values

Non-zero unsigned integer indicating the size in bytes to be used to evaluate when the log file needs to be rotated.

NOTE

- Default value of this property is 52428800 bytes (~ 50 MB) and should be changed as necessary.
 - This property is not used if the [Audit Log Local File Rotate Flag](#) is set to `false`
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

```
com.sun.am.policy.amAgentLog.local.file.rotate.size = 52428800
```

See Also

- [Audit Log Level](#)

- [Audit Log Disposition](#)
- [Audit Log Local File Name](#)
- [Audit Log Local File Rotate Flag](#)
- [Audit Log Remote File Name](#)

Audit Log Remote File Name

Property

`com.sun.am.policy.amAgentLog.remote.file`

Description

This property specifies the remote file name to be used by the agent for capturing audit messages on the remote system where the Sun ONE Identity Server Log service is available.

Valid Values

A String representing the file name (without a path) that will be used by the Sun ONE Identity Server Log service on the remote system to capture the audit messages sent by the agent.

NOTE

- If the file name specified by this property is already in use by another entity in the Sun ONE Identity Server network environment, the audit messages sent by this agent will be captured in the same file.
 - Invalid value specified for this property can result in the audit messages being lost by the system, which may not be available in the Local audit log file as well.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

`NONE, SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL`

Example

`com.sun.am.policy.amAgentLog.remote.file = myAgentLog`

See Also

- [Audit Log Level](#)
- [Audit Log Disposition](#)
- [Audit Log Local File Name](#)
- [Audit Log Local File Rotate Flag](#)
- [Audit Log Local File Rotation Size](#)

Bypass Principal List

Property

`com.sun.am.policy.amRealm.bypass.auth.principalList`

Description

This property specifies a list of principals that are bypassed by the agent for authentication and search purposes.

Valid Values

As set by the agent installation program. Any other value will be invalid.

NOTE

- This property is only used for the BEA WebLogic Server 6.1 SP2, BEA WebLogic Server 7.0 SP2, and BEA WebLogic Server 8.1 agents.
 - Invalid or inappropriate value specified for this property can lead to the system becoming inaccessible for some or all users.
 - This property must conform to the rules and format of LIST construct as described in the [General Notes on the Agent Configuration File](#) section.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amRealm.bypass.auth.principalList[0] = system
```

See Also

[General Notes on the Agent Configuration File](#)

PeopleSoft User Mapping

Property

`com.sun.am.policy.amPeopleSoft.user.mapping`

Description

This property specifies the way PeopleSoft user is mapped to Sun ONE Identity Server user.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

The valid value for this key can be one of the three strings: `USE_DN`, `LDAP`, or `HTTP_HEADER`. Any other value will be treated as an incorrect entry and the agent will default to `USE_DN`.

Mode `USE_DN`

In this mode of user mapping, the user ID from SSO token is used as the mapping attribute. user ID of the Sun ONE Identity Server user should be the same as that of the PeopleSoft user.

NOTE The following property is not used in this mode:
`com.sun.am.policy.amPeopleSoft.user.attribute.name`

Mode `LDAP`

In this mode of user mapping, PeopleSoft user is set as an LDAP attribute in Sun ONE Identity Server's user profile. Name of the LDAP attribute should be specified in the following property:

`com.sun.am.policy.amPeopleSoft.user.attribute.name`

Mode `HTTP_HEADER`

In this mode, HTTP header contains the PeopleSoft user id. The name of the LDAP attribute should be specified in the following property:

```
com.sun.am.policy.amPeopleSoft.user.attribute.name
```

Hot-Swap Enabled

No

Example

```
com.sun.am.policy.amPeopleSoft.user.mapping = USE_DN
```

See Also

[User Attribute Containing PeopleSoft User](#)

User Attribute Containing PeopleSoft User

Property

```
com.sun.am.policy.amPeopleSoft.user.attribute.name
```

Description

This property specifies the name of the attribute which contains the PeopleSoft user ID.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

In the LDAP mode of user mapping, it should be a valid LDAP attribute name and in the HTTP_HEADER mode of user mapping, it should be the name of HTTP Header containing PeopleSoft user ID.

NOTE Valid but inappropriate value of this property may lead to the application being unreachable by the users.

Hot-Swap Enabled

No

Examples

```
com.sun.am.policy.amPeopleSoft.user.attribute.name = employeenumber
```

```
com.sun.am.policy.amPeopleSoft.user.attribute.name = CUSTOM-HEADER
```

See Also:

[PeopleSoft User Mapping](#)

Validate SSO Token in PeopleCode flag

Property

```
com.sun.am.policy.amPeopleSoft.peoplecode.validate.sso
```

Description

This property specifies if the validation of SSO token should be done in PeopleSoft application server for extra security.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

`true`: indicates that the validation will be done

`false`: indicates that no validation will be performed

NOTE The default value of this property is `true` and should be changed as necessary.

Hot-Swap Enabled

No

Example

```
com.sun.am.policy.amPeopleSoft.peoplecode.validate.sso = true
```

URL Decode SSO Token Flag

Property

```
com.sun.am.policy.amPeopleSoft.sstoken.urldecode
```

Description

This property indicates if the SSO Token needs to be URL decoded by the agent before it is used in PeopleCode.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

`true`: indicates that the token should be URL Decoded.

`false`: indicates that the token should not be URL Decoded.

NOTE Valid but inappropriate value of this property may lead to the application being unreachable by the users.

Hot-Swap Enabled

No

Example

```
com.sun.am.policy.amPeopleSoft.sstoken.urldecode = true
```

Authentication Module

Property

```
com.sun.am.policy.amPeopleSoft.auth.module
```

Description

This property specifies the authentication module to be used for ensuring secure access to the protected resources.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

A string representing the name of the authentication module to be used which can be set as the URI parameter when redirecting the user to Sun ONE Identity Server Authentication Service.

NOTE The value of this property is set during installation and must not be changed unless absolutely necessary.

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amPeopleSoft.auth.module = LDAP
```

Goto URL

Property

```
com.sun.am.policy.amPeopleSoft.goto.url
```

Description

This property indicates the destination address for users who have successfully authenticated using the Sun ONE Identity Server Authentication service.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

A string representing the complete and fully qualified URL of the PeopleSoft application where the users will be directed after successful sign on.

-
- NOTE**
- The value of this property is set during installation and must not be changed unless absolutely necessary.
 - Please ensure that the value of this property is a valid entry point into the system. Failing to do so can result in the system becoming inaccessible.
-

Hot-Swap Enabled

Yes

Examples

```
com.sun.am.policy.amPeopleSoft.goto.url =  
http://www.mycompany.com:8001/servlets/iclientservlet/peoplesoft8/?cmd=sta  
rt
```

```
com.sun.am.policy.amPeopleSoft.goto.url =  
http://www.mycompany.com:8001/ps/ps/EMPLOYEE/ERP/h/?tab=DEFAULT
```

```
com.sun.am.policy.amPeopleSoft.goto.url =  
http://www.mycompany.com:8001/ps/ps/EMPLOYEE/HRMS/h/?tab=DEFAULT
```

Display Resource Root

Property

```
com.sun.am.policy.amPeopleSoft.display.resource.root
```

Description

This property specifies the directory where all the files that can be displayed to the user are available to the agent.

NOTE This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

A string representing the complete path to the directory where the files that can be displayed are kept.

NOTE

- If the value of this property is incorrect, the agent will not be able to display the necessary content to the user and would therefore default to redirecting the user to the Sun ONE Identity Server Authentication Service for login.
 - Any file that is displayed to the user from this directory is parsed by the agent before display and all occurrences of AUTH_LOGIN_URL are replaced by the actual login URL. If no occurrences of this string are found, no replacement is done.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amPeopleSoft.display.resource.root = /www/public_html
```

Default Display Resource File Name

Property

```
com.sun.am.policy.amPeopleSoft.display.resource.default
```

Description

This property specifies the name of the default HTML file that the agent will display to the user if the query parameter `cmd=display` is passed in the request.

NOTE

This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

A string representing the name of the HTML file that must be displayed to the user when `cmd=display` is passed to the `AgentAuthenticatorServlet`. This file should exist in the directory specified in the `Display Resource Root` property and should be readable by the WebLogic Server process.

NOTE

- If this file is not available for any reason, the `AgentAuthenticatorServlet` will default to redirecting the user to the Login URL instead.
 - The `AgentAuthenticatorServlet` will try to replace all occurrences of the string `AUTH_LOGIN_URL` with the appropriate login URL. If the contents of this file do not have any occurrences of this string, no replacement will be done.
 - If another query parameter of the form `resource=name` is passed, the `AgentAuthenticatorServlet` will ignore this property and instead use the following property which specifies the Display Resource Map to determine the file name of the file to be displayed to the user.
 - This property is set during the installation and must not be replaced unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amPeopleSoft.display.resource.default = display.html
```

Display Resource Map

Property

```
com.sun.am.policy.amPeopleSoft.display.resource.map
```

Description

This property specifies a map which relates various names that can be mapped to actual HTML files in the Display Resource Root directory. When the `AgentAuthenticatorServlet` is invoked with the query parameters `cmd=display` and `resource=name`, the agent uses this map to identify the file name associated with the passed `name` value.

When found, the `AgentAuthenticatorServlet` will replace all occurrences of the string `AUTH_LOGIN_URL` in the text of this file and display the resulting text to the end user.

NOTE

This property is used only for the policy agent for PeopleSoft 8.3/8.4/8.8.

Valid Values

Valid values must comply with the syntax of this property. See Note and Example.

The specified file name in this map must correspond to a real HTML file that exists in the Display Resource Root directory and this file should be readable by the agent process.

NOTE

- The syntax of this property is:
`com.sun.am.policy.amPeopleSoft.display.resource.map[name]=f
ilename`, where *name* is the resource parameter value passed into the `AgentAuthenticatorServlet` and *filename* is the actual name of the HTML file that must be processed for display.
 - Any number of such properties may be specified as long as they are valid properties conforming to the requirements stated above.
-

Hot-Swap Enabled

Yes

Examples

```
com.sun.am.policy.amPeopleSoft.display.resource.map[logout] = logout.html
```

```
com.sun.am.policy.amPeopleSoft.display.resource.map[error] = error.html
```

CDSSO Enable Flag

Property

```
com.sun.am.policy.amFilter.cdsso.enabled
```

Description

This property specifies if Cross Domain Single Sign On (CDSSO) is enabled for this agent. This property should be enabled if the agent resides in a DNS domain which is different from the Sun ONE Identity Server domain.

Valid Values

- `true`: indicates that CDSSO has been enabled
- `false`: indicates that CDSSO has been disabled

NOTE By default, the value of this property is set to `false`. In order to achieve SSO when an agent is installed in a domain different from the Sun ONE Identity Server domain, the property must be enabled.

Hot-Swap Enabled

Yes

See Also

- [CDSSO Intermediate Redirect URI](#)
- [CDC Servlet URL](#)
- [CDSSO Request Cookie Name](#)
- [CDSSO Liberty Assertion Validity Clock Skew Factor](#)

Example

```
com.sun.am.policy.amFilter.cdsso.enabled = true
```

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

CDC Servlet URL

Property

```
com.sun.am.policy.amFilter.cdsso.cdcServletURL[0]
```

Description

This property specifies the CDC Servlet URLs to be used by the agent to redirect incoming users without sufficient credentials to the Sun ONE Identity Server authentication service when CDSSO is enabled.

Valid Values

A string that represents the complete URL to be used as the redirect URL in order to send users without sufficient credentials to Sun ONE Identity Server authentication service.

NOTE This property is set during agent installation and must not be changed unless absolutely necessary.

This property must conform to the rules and format of LIST construct as described in the section [General Notes on the Agent Configuration File](#).

Hot-Swap Enabled

Yes

See Also

- [CDSSO Enable Flag](#)
- [CDSSO Intermediate Redirect URI](#)
- [CDSSO Request Cookie Name](#)
- [CDSSO Liberty Assertion Validity Clock Skew Factor](#)

Example

```
com.sun.am.policy.amFilter.cdsso.cdcServletURL[0] =  
http://www.mycompany.com:58080/cdcervlet
```

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

CDSSO Intermediate Redirect URI

Property

```
com.sun.am.policy.amFilter.cdsso.redirect.uri
```

Description

This property specifies the URI to which the CDSSO Controller will redirect the authentication responses. A valid value must be supplied if the CDSSO is enabled. By default, the value will be `sunwCDSSORedirectURI`.

Valid Values

A valid URI that belongs to the primary application on this server.

-
- NOTE**
- If an invalid value is specified for this property, it can result in CDSSO not functioning and the application becoming inaccessible.
 - This property comes into effect only if CDSSO has been enabled.
 - This property is set during agent installation to `sunwCDSSORedirectURI` by default and must not be changed unless absolutely necessary.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amFilter.cdsso.redirect.uri = /sunwCDSSORedirectURI
```

See Also

- [CDSSO Enable Flag](#)
- [CDC Servlet URL](#)
- [CDSSO Request Cookie Name](#)
- [CDSSO Liberty Assertion Validity Clock Skew Factor](#)

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

CDSSO Request Cookie Name

Property

```
com.sun.am.policy.amFilter.cdsso.cookie.name
```

Description

This property specifies the name of the cookie that will be used to keep track of the user requested resource when CDSSO is enabled.

Valid Values

A string that represents the name of the cookie to be issued by the agent in order to keep track of the user requested resource when CDSSO is enabled.

NOTE This property is set during agent installation and need not be changed unless absolutely necessary.

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amFilter.cdsso.cookie.name = amFilterCDSSORequest
```

See Also

- [CDSSO Enable Flag](#)
- [CDC Servlet URL](#)
- [CDSSO Intermediate Redirect URI](#)
- [CDSSO Liberty Assertion Validity Clock Skew Factor](#)

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

CDSSO Liberty Assertion Validity Clock Skew Factor

Property

```
com.sun.am.policy.amFilter.cdsso.clock.skew
```

Description

This property specifies a skew factor (time in seconds) to be used by the agent to determine the validity of the CDSSO AuthnResponse assertion. This property provides a means to avoid request denials caused by expired assertions, which are related to network latency.

Valid Values

Unsigned integer value, including 0, which indicates the amount of time in seconds that will be used by the agent to create an interval window based on the system time of the agents' host machine for validating the assertions.

NOTE

- This property is set during agent installation and can be modified to accommodate the network latency of the host system.
 - This property comes into effect only if CDSSO has been enabled.
-

Hot-Swap Enabled

Yes

See Also

- [CDSSO Enable Flag](#)
- [CDC Servlet URL](#)
- [CDSSO Intermediate Redirect URI](#)
- [CDSSO Request Cookie Name](#)

Example

```
com.sun.am.policy.amFilter.cdsso.clock.skew = 90
```

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Login Form Error List

Property

```
com.sun.am.policy.amFilter.login.errorList
```

Description

This property specifies the list of URIs that represent the `form-error-page` as specified in the `form-login-config` elements of the protected application's deployment descriptors (`web.xml`) along with the context path of the deployed application. This property is used by the agent to intercept error page requests and take appropriate action.

Valid Values

- Values of this property must comply with the syntax of this property.
- Values should be as specified in the `form-error-page` element of `web.xml` of the deployed application along with the context path of the application.

NOTE

- This property is applicable only to the agents for Oracle 9iAS R2, Oracle 10g and Tomcat Server 4.1.27.
 - This property must conform to the rules and format of LIST construct as described in the section [General Notes on the Agent Configuration File](#).
 - The value of `form-error-page` in `web.xml` of deployed application is only a part URI, which must be preceded with the context path of the application. For instance, if the application context path is `/Portal` and the value of `form-error-page` is `/jsp/error.jsp`, then the actual value for this configuration becomes `/Portal/jsp/error.jsp`.
 - Failure to set this property correctly can lead to the malfunctioning of the agent resulting in the application or portions thereof becoming inaccessible.
-

Hot-Swap Enabled

Yes

Examples

```
com.sun.am.policy.amFilter.login.errorList[0] = /Portal/jsp/error.jsp
```

```
com.sun.am.policy.amFilter.login.errorList[1] = /BankApp/LoginError.html
```

See Also

- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [J2EE Logout Handler](#)
- [Form Login Use Internal Flag](#)

- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)
- [Default Referer Map for Form Login](#)

Applicable Modes of Operation

J2EE_POLICY, ALL

Agent Operation Mode Map

Property

`com.sun.am.policy.amFilter.mode.map`

Description

This property allows the agent filter to be configured for different modes of operation for different deployed applications. The value of operation mode specified for a given application in this map takes precedence over the agent operation mode.

Valid Values

An appropriate agent operation mode for a given application protected by this agent.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - This property is set during agent installation, if needed, and should not be changed unless absolutely necessary.
 - The names used in this map are context path names of the application without the leading '/' character.
 - The values specified in this map can be any valid operation modes of the agent. For details on various operation modes, please refer to the section [Agent Filter Modes](#).
 - This property takes precedence over the agent operation mode specified by the property `com.sun.am.policy.amFilter.mode`.
 - For applications that are not explicitly added to this map, the operation mode is determined by the property `com.sun.am.policy.amFilter.mode`.
 - This property is applicable only to the agents for Oracle 9iAS R2, Oracle 10g and Tomcat Server 4.1.27.
-

Hot-Swap Enabled

No

See Also

- [Agent Filter Mode](#)
- [J2EE Authentication Handler](#)
- [J2EE Logout Handler](#)
- [Login Form List](#)
- [Form Login Use Internal Flag](#)
- [Form Login Content File Name](#)
- [Preserve Referer for Form Login Flag](#)

Examples

```
com.sun.am.policy.amFilter.mode.map[Portal]=J2EE_POLICY
```

```
com.sun.am.policy.amFilter.mode.map[BankApp]=ALL
```

Applicable Modes of Operation

J2EE_POLICY, ALL

Enable Filtered Roles

Property

`com.sun.am.policy.amRealm.filtered.roles.enable`

Description

This property specifies if the agent should enable the use of filtered roles when evaluating role-to-principal mappings. By default, the agent uses static roles as defined in Sun ONE Identity Server. This behavior can be changed by setting the value of this property as `true`. When set to `true`, the agent will use both the static as well as the filtered roles to evaluate role-to-principal mappings.

Valid Values

`true`: indicates that the agent will use filtered roles for evaluating role-to-principal mappings.

`false`: indicates that the agent will not use filtered roles for evaluating role-to-principal mappings.

NOTE

- The value of this property is set during agent installation and may be changed as necessary.
 - It is recommended that if the application does not rely on the use of filtered roles to achieve role-to-principal mapping, the value of this property should be set to `false`.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amRealm.filtered.roles.enable = true
```

SAP User Mapping

Property

`com.sun.am.policy.amSAP.user.mapping`

Description

This property specifies the way an SAP Enterprise Portal user is mapped to an Identity Server user.

Valid Values

The valid value for this key can be one of the following three strings: `USE_DN`, `LDAP`, and `HTTP_HEADER`. Any other value will be treated as invalid and the agent will default to `USE_DN`.

Mode USE_DN

In this mode, the user ID from SSO token is used as the mapping attribute. User ID of the Identity Server user should be same as that of the SAP user. Note that in this mode, the property `com.sun.am.policy.amSAP.user.attribute.name` is not used.

Mode LDAP

In this mode, the SAP user name is set as an LDAP attribute in Identity Server's user profile. Name of the LDAP attribute should be specified in the property `com.sun.am.policy.amSAP.user.attribute.name`.

Mode HTTP_HEADER

In this mode, the HTTP header contains the SAP user name. Name of the HTTP header should be specified in the property `com.sun.am.policy.amSAP.user.attribute.name`.

NOTE

- When the mapping is set to `HTTP_HEADER`, the supplied `HTTP_HEADER` value must be provided with the incoming request. Headers injected by the agent itself may not be used as valid mapping configuration setting.
 - This property is set to `USE_DN` during agent installation and should be changed as necessary.
 - This property is used only for the policy agent for SAP Enterprise Portal 6.0 SP2.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amSAP.user.mapping = USE_DN
```

See Also

[User Attribute Containing SAP User-ID](#)

User Attribute Containing SAP User-ID

Property

```
com.sun.am.policy.amSAP.user.attribute.name
```

Description

This property specifies the name of the attribute that contains the SAP user-id. Depending on the mode in which the agent enforces user mapping, this name could refer to an LDAP Attribute or to a HTTP Header field name.

Valid Values

In the LDAP mode of user mapping, the value should be a valid LDAP attribute name and in the HTTP_HEADER mode of user mapping, it should be the name of the HTTP Header that contains the SAP user-id.

NOTE

- Valid but inappropriate value of this property may lead to the application being unreachable by the users.
 - This property is set to the value `employeenumber` during agent installation and should be changed as necessary.
 - This property is used only by the policy agent for SAP Enterprise Portal 6.0 SP2.
-

Hot-Swap Enabled

No

Examples

```
com.sun.am.policy.amSAP.user.attribute.name = employeenumber
```

```
com.sun.am.policy.amSAP.user.attribute.name = CUSTOM-HEADER
```

See Also

[SAP User Mapping](#)

Logon Frontend Tracking Cookie Name

Property

```
com.sun.am.policy.amSAP.lfe.cookie.name
```

Description

This property specifies the name of the cookie used by the agent logon frontend component to track the user session. This component is used within the iView Runtime Environment of SAP Enterprise Portal for processing login requests.

Valid Values

A string that represents the name of the cookie to be issued by the agent in order to track the user session and synchronize it with the Identity Server session.

NOTE

- This property is set during agent installation and must not be changed unless absolutely necessary.
 - This property is used only by the agent for SAP Enterprise Portal 6.0 SP2.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amSAP.lfe.cookie.name = amLFEPParam
```

Goto URL

Property

`com.sun.am.policy.amSAP.goto.url`

Description

This property specifies the destination URL for users who have successfully authenticated using the Sun ONE Identity Server authentication service. However, this property is only used when the users that have been allowed access to the SAP Enterprise Portal click the SAP Enterprise Portal's Logout link or button.

Valid Values

A string representing the complete and fully qualified URL of the destination where the users will be sent after successful logon.

NOTE

- The value of this property is set during installation and should not be changed unless absolutely necessary.
 - The value of this property must be set to a valid entry point into the system. Failing to do so can result in the system becoming inaccessible.
 - This property is used when a user clicks the logout link or button in SAP Enterprise Portal environment that is protected by the agent. It is not used to direct incoming users that have not yet established an SAP Enterprise Portal session.
 - This property is specific to the agent for SAP Enterprise Portal 6.0 SP2.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amSAP.goto.url =  
http://www.mycompany.com:50000/irj/index.html
```

Error Content File Name

Property

`com.sun.am.policy.amSAP.error.filename`

Description

This property specifies the name of the file that has the necessary content needed to handle error situations. When an error is encountered while evaluating the request, the agent displays the contents of this file after substituting the appropriate logout link in it.

Valid Values

Name of a file that exists in the *Agent_install_dir/locale* directory or the complete path to a file that exists on the system and is readable by the server process on which the agent is installed.

NOTE

- The file being pointed to by this property must have read permissions for the server process on which the agent is installed.
 - If the contents of this file are modified or if this property is set to point to another file instead of the one provided by the agent, it must have the string `am.sap.logout.url` specified appropriately. This string is replaced by the agent with the actual logout URL before the contents of this file are displayed to the end user.
 - This property is set during agent installation and should not be changed unless absolutely necessary.
 - This property is specific to the agent for SAP Enterprise Portal 6.0 SP2.
-

Hot-Swap Enabled

Yes

Example

```
com.sun.am.policy.amSAP.error.filename = CustomErrorContent.txt
```

User Mapping

Property

```
com.sun.am.policy.user.mapping
```

Description

This property specifies the way a local user is mapped to an Identity Server user. Local user refers to the way the user is identified by the platform on which the agent is running.

Valid Values

The valid value for this key can be one of the following three strings: `USE_DN`, `LDAP`, or `HTTP_HEADER`. Any other value will be treated as invalid and the agent will default to `USE_DN`.

Mode USE_DN

In this mode, the user ID from the SSO token is used as the mapping attribute. The user ID of the Identity Server user should be the same as that of the local user. Note that in this mode, the following property is not used:

```
com.sun.am.policy.user.attribute.name
```

Mode LDAP

In this mode, the local user name is set as an LDAP attribute in Identity Server's user profile. The name of the LDAP attribute should be specified in the following property:

```
com.sun.am.policy.user.attribute.name.
```

CAUTION Ensure that the `LDAP` attribute contains one and only one value. Multiple values or a null value for the `LDAP` attribute are unrecognizable by the agent, which causes the agent to function improperly

Mode HTTP_HEADER

In this mode, the HTTP header contains the local user name. The name of the HTTP header should be specified in the following property:

```
com.sun.am.policy.user.attribute.name.
```

CAUTION Ensure that the `HTTP_HEADER` attribute contains one and only one value. Multiple values or a null value for the `HTTP_HEADER` attribute are unrecognizable by the agent, which causes the agent to function improperly

NOTE

- When the mapping is set to `HTTP_HEADER`, the supplied HTTP header value must be provided with the incoming request. Headers injected by the agent itself can not be used as valid mapping configuration settings.
- This property is set to `USE_DN` during agent installation and should be changed as necessary.

Hot-Swap Enabled

Yes

Applicable Modes of Operation:

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.user.mapping = USE_DN
```

See Also

[User Attribute Containing Agent User](#)

User Mapping Mode

Property

```
com.sun.am.policy.user.mapping.mode
```

Description

This property specifies the mode through which the portal user is mapped to Identity Server user.

Valid Values

The valid value for this key can be one of the three strings: `USE_DN`, `LDAP`, or `HTTP_HEADER`

NOTE Valid but inappropriate values for this property could lead to the application being unreachable by users.

Mode `USE_DN`

In this mode of user mapping, the user ID from the SSO token is used as the mapping attribute. The user ID of the Identity Server user should be the same as that of the WebLogic 8.1 SP2/SP3 Server/Portal user. By default the agent is running in `USE_DN` mode.

Mode LDAP

In this mode of user mapping, WebLogic 8.1 SP2/SP3 Server/Portal user is set as an LDAP attribute in Identity Server's user profile. The name of the LDAP attribute should be specified in the following property:

```
com.sun.am.policy.user.attribute.name
```

Mode HTTP_HEADER

In this mode, the HTTP header contains the WebLogic 8.1 SP2/SP3 Server/Portal user ID. The name of the LDAP attribute should be specified in the following property:

```
com.sun.am.policy.user.attribute.name
```

Hot-Swap Enabled

Yes

Applicable Modes of Operation:

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Example

```
com.sun.am.policy.user.mapping = USE_DN
```

See Also

[User Attribute Containing Agent User](#)

User Attribute Containing Agent User

Property

```
com.sun.am.policy.user.attribute.name
```

Description

This property specifies the name of the attribute which contains the user ID.

Valid Values

In the LDAP mode of user mapping, a valid LDAP attribute name should be used. In the HTTP header mode of user mapping, the name of the HTTP header containing the user ID should be used.

NOTE

- Valid but inappropriate value of this property may lead to the application being unreachable by the users.
- This property is set to `employeenumber` during agent installation. For example:
`com.sun.am.policy.user.attribute.name = employeenumber`

If necessary, change the setting to another value.

NOTE A valid but inappropriate value of this property may lead to the application being unreachable by users.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

SSO_ONLY, URL_POLICY, J2EE_POLICY, ALL

Examples

```
com.sun.am.policy.user.attribute.name = employeenumber
```

```
com.sun.am.policy.user.attribute.name = CUSTOM-HEADER
```

See Also

- [User Mapping Mode](#)
- [User Mapping](#)

Logout Parameter Map

Property

```
com.sun.am.policy.amFilter.logout.request.param.map
```

Description

This property specifies the application specific logout parameter value used by the agent to detect the need to invalidate the Application Server and Identity Server user. The logout parameter will be first searched in the HTTP request body depending on the value of the following parameter:

[HTTP Request Body Introspect Flag](#)

`com.sun.am.policy.amFilter.logout.introspect.request.enable`

Valid Values

If the value of this property is set to *false*, the Logout parameter will be searched in the HTTP request URI query string only. By default the `introspect.request.enable` property is set to *true*.

NOTE

- This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - The names used in this map are context path names of the application without the leading '/' character.
 - - Either the presence of an entry in Logout Parameter Map or the presence of an entry in Logout URI Map will trigger the need to invalidate the user.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

If the agent is trying to protect a portal application with the context URI of `/portaApp` and the logout event for `/portalApp` adds a request parameter `logout` in the HTTP request, the Administrator needs to create the following entry for Logout Parameter Map:

```
com.sun.am.policy.amFilter.logout.request.param.map[portaApp]=logout
```

See Also

- [Logout URI Map](#)
- [HTTP Request Body Introspect Flag](#)

- [Application Local Logout Handler Map](#)

Logout URI Map

Property

`com.sun.am.policy.amFilter.logout.uri.map`

Description

This property specifies the application specific logout URI value used by the agent to detect the need to destroy/invalidate the Application Server and Identity Server sessions. If the request URI matches the logout URI specified in the map, the agent will proceed to invalidate the Application Server and Identity Server sessions.

Valid Values

A string that represents a valid URI.

NOTE

- This property must conform to the rules and format of MAP construct as described in ["Map Constructs in the Configuration File" on page 155](#).
 - The names used in this map are context path names of the application without the leading '/' character.
 - Either the presence of an entry in Logout URI Map or the presence of an entry in Logout Parameter Map will trigger the need to invalidate the user.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.logout.uri.map[portalApp]=/portalApp/logoutURL.html
```

See Also

[Logout Parameter Map](#)

Application Authentication Handler Map

Property

`com.sun.am.policy.amFilter.mapped.authentication.handler`

Description

This property specifies the application specific local authentication handler used by the agent to validate the user credentials with the application container. All applications protected by the agent can have an authentication handler.

If a specific application has an authentication handler; it will be loaded by the agent. If the agent fails to load the handler for an application that has an authentication handler specified in the map, the agent can malfunction, resulting in the application or portions thereof becoming inaccessible. If the application does not have an authentication handler specified in the map, then user credentials will be validated using the default authentication handler specified in the following property:

[J2EE Authentication Handler](#)

`com.sun.am.policy.amFilter.j2ee.auth.handler`

Valid Values

A valid local authentication handler class.

NOTE

- The value of this property is set during installation and should not be changed unless absolutely necessary.
 - This property must conform to the rules and format of MAP construct as described in [“Map Constructs in the Configuration File” on page 155](#).
 - The names used in this map are context path names of the application without the leading ‘/’ character.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.mapped.authentication.handler[PortalApp] =  
AuthenticationHandlerClassForPortal
```

```
com.sun.am.policy.amFilter.mapped.authentication.handler[BankApp] =  
AuthenticationHandlerClassForBank
```

See Also

[J2EE Authentication Handler](#)

Verification Handler Map

Property

```
com.sun.am.policy.amRealm.verification.handler.map
```

Description

This property specifies the application specific local verification handler used by the agent to validate the user credentials with the local repository. All applications protected by the agent can have a verification handler. If a specific application has a verification handler, it is loaded by the agent. If the agent fails to load the handler for an application that has a verification handler specified in the map, the agent can malfunction, resulting in the application or portions thereof becoming inaccessible. If the application does not have a verification handler specified in the map, no verification process for the user ID is invoked.

Valid Values

A valid local verification handler class.

NOTE

- This property must conform to the rules and format of MAP construct as described in ["Map Constructs in the Configuration File" on page 155](#).
 - The names used in this map are context path names of the application without the leading '/' character.
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amRealm.verification.handler.map[PortalApp] =  
VerificationHandlerClassForPortalApp
```

Global Verification Handler

Property

com.sun.am.policy.amRealm.verification.handler

Description

This property specifies the global or default verification handler used by the agent to validate the user credentials with the local repository. All applications protected by the agent will use this handler if a match for a local verification handler is not found.

Valid Values

A valid global verification handler class.

NOTE This property should not be changed after installation.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

See Also:

[Verification Handler Map](#)

HTTP Request Body Introspect Flag

Property

`com.sun.am.policy.amFilter.logout.introspect.request.enable`

Description

This property indicates whether or not the Logout parameter specified for the application in the following property for the application URI can be searched in the HTTP request body:

[Logout Parameter Map](#)

`com.sun.am.policy.amFilter.logout.request.param.map`

Valid Values

A value of *true* indicates that the HTTP request can be searched for the Logout parameter while a value of *false* indicates that the HTTP request cannot be searched for the Logout parameter.

NOTE This property should not be changed after installation.

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.logout.introspect.request.enable = false
```

See Also:

- [Logout URI Map](#)
- [Application Local Logout Handler Map](#)

Application Local Logout Handler Map

Property

`com.sun.am.policy.amFilter.logout.application.handler.map`

Description

This property specifies the application specific Local Logout Handler parameter used by the agent to invalidate the HTTP session in the Application server. All applications protected by the agent can have a Local Logout Handler. If a specific application has a Local Logout Handler, it will be loaded by the agent. If the agent fails to load the application specific Local Logout Handler, the agent can malfunction, resulting in the application or portions thereof becoming inaccessible.

Valid Values

NOTE

- The value of this property is set during installation and should not be changed unless absolutely necessary.
 - This property must conform to the rules and format of MAP construct as described in ["Map Constructs in the Configuration File" on page 155](#).
 - The names used in this map are context path names of the application without the leading '/' character
-

Hot-Swap Enabled

Yes

Applicable Modes of Operation

J2EE_POLICY, ALL

Example

```
com.sun.am.policy.amFilter.logout.application.handler.map[sampleApp] =  
LocalLogoutHandlerForSampleApp
```

See Also:

- [Logout URI Map](#)
- [Logout Parameter Map](#)

Tools and APIs

J2EE Policy Agents provide tools and APIs that can be used to better manage your system and create security-aware applications.

This chapter describes the tools as well as APIs provided by the agents. Topics in this chapter include:

- [Agent Tools](#)
- [Agent APIs](#)

Agent Tools

The agent tools allow you to encrypt plain text strings, which may be required when changing various passwords and also allow you to configure the agent for new application server instances where supported.

The agent tools can be used by invoking the tool script provided by the agent. This script is located in the directory `bin` directory of agent installation. On Solaris, HP-UX and AIX platforms, this script is called `agentadmin` and on Windows platform it is called `agentadmin.bat`.

When invoked at a shell prompt from the `bin` directory of the agent installation, this script displays the usage of the tools as follows:

```
# ./agentadmin
Usage: agentadmin -option <arguments> [-debug],
where option => [encrypt | configure | unconfigure]
Example: agentadmin -encrypt <plain text>
Example: agentadmin -configure -debug
Example: agentadmin -unconfigure -debug
```

On Windows platform, this command can be invoked as follows:

```
C:\Sun\IdentityServer\j2ee_agents\bin> agentadmin
Usage: agentadmin -option <arguments> [-debug],
where option => [encrypt | configure | unconfigure]
Example: agentadmin -encrypt <plain text>
Example: agentadmin -configure -debug
Example: agentadmin -unconfigure -debug
```

Currently the tools support the following functionality: encryption, configuration and unconfiguration.

NOTE The configure and unconfigure options of the agent tools script are not supported for IBM WebSphere Application Server 5.0/5.1 agent. In order to manually configure a new instance of IBM WebSphere Application Server 5.0/5.1, you can refer to the steps described in the section [“Agent for IBM WebSphere Application Server 5.0/5.1 Instance” on page 280](#) and to unconfigure an instance, refer to the section [Agent for IBM WebSphere Application Server 5.0/5.1 Instance](#).

Using Tools to Encrypt Strings

In order to encrypt a string, simply invoke the agent tools with the `-encrypt` option as shown here and provide the necessary string that needs to be encrypted. The agent tools will then display the text of the encrypted string.

```
#./agentadmin -encrypt <string to encrypt>
Encrypted Value => <encrypted value of the string>
```

On Windows platform, use the following command:

```
C:\Sun\IdentityServer\j2ee_agents\bin> agentadmin -encrypt <string
to encrypt>
Encrypted Value => <encrypted value of the string>
```

Configuring the Agent for an Application Server Instance

Once you have installed a policy agent for a particular application server, the agent installation program will not allow you to install the same agent again unless the previous installation has been completely removed from your system. In order to overcome this limitation, you may use the agent tools to configure the agent for another application server instance.

-
- NOTE**
- The configure and unconfigure options of the agent tool are not supported for IBM WebSphere Application Server 5.0/5.1 agent. In order to manually configure the agent for a new instance of IBM WebSphere Application Server 5.0/5.1, you can refer to the steps described in the section [Agent for IBM WebSphere Application Server 5.0/5.1 Instance](#) and to unconfigure an instance, refer to the section [Agent for IBM WebSphere Application Server 5.0/5.1 Instance](#).
 - For every application deployed on any instance of the application server that is protected by the agent, you must ensure that the agent filter has been added and appropriate changes to the deployment descriptors have been made to enable web-tier declarative security as needed. Refer to the section on [Enabling Web-Tier Declarative Security](#) for further details.
-

The following steps describe how the agent can be configured for a new instance of an application server using the agent tools.

These steps are divided into some common steps that apply to all the agents, followed by steps specific to an agent. After performing the steps outlined in the following section [Common Configuration Steps](#), you may jump to the section relevant to the agent you want to configure.

-
- NOTE** At any time when using the agent tools to configure a new server instance, you may select the default value shown in brackets [] by pressing the Enter key at the prompt.
-

Common Configuration Steps

The following steps are common to all the agents. After you have completed these steps, you must perform the configuration tasks specific to your agent, which are covered in the next sections.

-
- NOTE** If you are configuring the agent for a server instance of Macromedia JRun 4, you must first stop the instance and then start the configuration.
-

1. From the `bin` directory of the agent installation, invoke the agent tool script as follows:

```
#!/agentadmin -configure
```

On Windows platform, use this command:

```
agent_install_dir\IdentityServer\j2ee_agents\bin> agentadmin -configure
```

2. At the following prompt, enter the protocol for the application server instance being configured.

```
Server Instance Preferred Protocol ('http' or 'https') [http]?:
```

3. At the following prompt, enter the port for the application server instance being configured.

```
Server Instance listening port [80]?:
```

4. At the following prompt, enter the context path for the primary application deployed on this instance of application server.

```
Primary Application Context Path [/]?:
```

5. Enter the value for access denied URI that will be used by the agent for this instance of application server to blocking unauthorized requests.

```
Access Denied URI []?:
```

You may leave this value empty, in which case the agent will use the HTTP status code 403 to respond to unauthorized access requests.

6. Enter the value for agent operation mode from the valid set of choices as displayed by the agent tools:

```
Agent instance Filter Mode ('NONE', 'SSO_ONLY', 'URL_POLICY', 'J2EE_POLICY', 'ALL') [ALL]?:
```

NOTE When the agent is configured for a new instance of the application server, the agent tools use the same user name and group name as specified during the actual installation to set the appropriate permissions for the debug and configuration directories. If this is not applicable, you must manually reset these permissions to the appropriate owner of the new application server instance process.

The information you have entered so far has been common to agents for all application servers that support the `configure` option. However, the next set of information is specific to each type of application server agent. Please jump to the relevant section to continue with the application server instance configuration.

- [Agent for Sun ONE Application Server 7.0](#)
- [Agents for BEA WebLogic Server \(All Versions\)](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1 Instance](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Macromedia JRun 4](#)
- [Agents for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

After you have completed the steps outlined in the [Common Configuration Steps](#) section, the agent tool will prompt you for information specific to the Sun ONE Application Server 7.0 instance that is being configured. Follow these steps:

1. At the following prompt, enter the location of the config directory for the application server instance that is being configured:

```
Enter the Sun ONE Application Server instance config Directory?:
```

2. At the following prompt, enter the port number where Sun ONE Application Server Administration Server is available. Make sure that the administration server is running when doing this configuration:

```
Enter Sun ONE Application Server Administrative Server port
[4848]?:
```

3. At the following prompt, enter the user name for the Administrator of Sun ONE Application Server.

```
Enter Sun ONE Application Server Administrative User Name
[admin]?:
```

4. At the following prompt, enter the password for the Administrator of Sun ONE Application Server:

```
Enter Sun ONE Application Server Administrative User Password?:
```

5. At the following prompt, re-enter the password for the Administrator of Sun ONE Application Server:

```
Reenter Sun ONE Application Server Administrative User Password?:
```

If the information entered is valid, the agent tools configure the Application Server instance. On completion of this task, prior to the program exiting, the tools display the following message indicating status:

```
Agent Configured Successfully for Application Server Instance.
```

Agents for BEA WebLogic Server (All Versions)

This section addresses the configuration of agents for any of the following:

- BEA WebLogic Server 6.1 SP2,
- BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1
- BEA WebLogic 8.1 SP2/SP3 Server/Portal

The agent tools will prompt you for information specific to the WebLogic server instance after you have completed the steps outlined in the [Common Configuration Steps](#) section.

At the following prompt, enter the location of the startup script that is used to launch this instance of WebLogic Server or Portal:

```
Enter the location of WebLogic Server Startup Script?:
```

If the information entered is valid, the agent tools proceed to configure the instance of the application server. When the configuration is complete, the tools display a message indicating the status of the task. The program then exits and this message is displayed:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE If the BEA WebLogic Server instance that has been configured is outside the domain in which the agent was installed, you must complete the post-installation tasks for the respective server instance as mentioned in the [Post-Installation Tasks](#) section in Chapter 2, [Installing the Agent](#).

Agent for IBM WebSphere Application Server 5.0/5.1 Instance

The agent tools do not support the `configure` option for IBM WebSphere Application Server 5.0/5.1 instances. However, the following list of steps allow you to manually configure an agent for a new instance of the application server:

1. Create a new directory for storing the agent configuration for the new instance of the application server under the agent configuration root directory. On Solaris platform, the agent configuration root directory is in the path `/etc/opt/SUNWam/j2ee_agents/am_was50_agent/config`. On other platforms, this directory is under `Agent_Install_Dir/IdentityServer/j2ee_agents/config` directory. Make sure that this is a unique directory that will not be used by any other instance of the application server other than the one being configured.

The rest of the steps will refer to this directory as the new configuration directory.

2. Under the new configuration directory, create a directory called `ums` for storing Sun ONE Identity Server's Directory server configuration file `serverconfig.xml`.
3. From the configuration directory of the previously installed agent, copy the files `AMAgent.properties` and `SSOConfig.properties` to the new configuration directory.
4. From the directory `config/ums` of the previously installed agent, copy the `serverconfig.xml` file to the `ums` directory created under the new configuration directory.
5. Create a new directory for storing the agent audit and debug logs for the new instance of the application server under the agent log root directory. On Solaris platform, the agent log root directory is in the path `/var/opt/SUNWam/j2ee_agents/am_was50_agent`. On other platforms, this directory is under `Agent_Install_dir/IdentityServer/j2ee_agents/logs` directory. Make sure that this is a unique directory that will not be used by any other instance of the application server other than the one being configured. The rest of the steps will refer to this directory as the new log directory.
6. Under the new log directory, create two directories named `audit` and `debug`.
7. Using the IBM WebSphere Application Server 5.0/5.1 Administrative Console, create a new instance of the application server as a clone of the instance for which the agent has been already installed. To do this, go to Servers and then Application Servers and click the New button. Now select the Existing application server radio button, enter a name for the new server instance and click Next.
8. Once the new server instance has been created, navigate to Advance Java virtual machine setting page for the new Server instance by following the links as Application Servers > <new server name> > Process Definition > Java Virtual Machine.

9. On this form, under the classpath properties, modify the classpath entry that points to the old instance's config directory and replace it with the new config directory created in step 1.
10. On the same form, under the Generic JVM Arguments, modify the existing arguments and replace the value of the parameter `-Dcom.iplanet.coreservices.configpath` with the complete path to the ums directory created under the new config directory in step 2. Also, replace the value of the parameter `-Djava.util.logging.config.file` with the complete path to the new `AMAgent.properties` file created in step 3.
11. Save your changes in the IBM WebSphere Application Server 5.0/5.1 Administration console.

12. Before restarting the server, modify the `AMAgent.properties` file created under the new config directory and set the values for the following properties:

```
com.sun.am.policy.amFilter.port.check.map
```

Set the port and protocol used by this instance of the application server.

```
com.sun.am.policy.amAgentLog.local.file
```

Set the path of a file that will reside in the audit directory created under the new log directory in step 6.

```
com.iplanet.services.debug.directory
```

Set the path of the debug directory created under the new log directory in step 6.

```
com.iplanet.am.notification.url
```

Set the notification URL with the appropriate port and protocol and make sure that the URI for this begins with the context path of the primary application that will be deployed on this instance.

```
com.sun.identity.agents.notification.url
```

Set the notification URL with the appropriate port and protocol and make sure that the URI for this begins with the context path of the primary application that will be deployed on this instance.

Refer to the section on [AMAgent.properties Reference](#) for details regarding these property keys.

13. Restart the WebSphere Application Server instance.

Agent for PeopleSoft 8.3/8.4/8.8

If you are configuring the agent for PeopleSoft 8.3/8.4/8.8, the agent tools will prompt you for information specific to the PeopleSoft application server instance after you have completed the steps outlined in the [Common Configuration Steps](#) section. Follow these steps:

1. At the following prompt, enter the PeopleSoft application version on which you are going to configure the agent.

```
Enter the PeopleSoft application version ('8.3' or '8.4' or
'8.8')      [8.3] ?:
```

2. At this prompt, enter the mode in which you want to deploy the agent.

```
Enter one of the Agent Deployment options ('redeploy' or
'proxy')   [redeploy] ?:
```

3. At this prompt, type `true` if you have PeopleSoft Application Server installed locally. Else, type `false`.

```
PeopleSoft Application Server installed locally ('true' or
'false')   [true] ?:
```

4. At this prompt, type `true` if you have BEA WebLogic Server installed locally. Else, type `false`.

```
WebLogic Server installed locally ('true' or 'false')
[true]?:
```

5. At this prompt, enter the full path to the directory where the PeopleSoft Application Server is installed. For example, `PS_HOME/appserv`

```
Enter PeopleSoft AppServer Directory
[/export/home/psft/appserv] ?:
```

6. At this prompt, enter the PeopleSoft domain that this agent will protect. For example, `HDMO`.

```
Enter PeopleSoft Domain Name      [HDMO] ?:
```

7. At this prompt, enter the full path to the WebLogic instance directory. For example, `PS_HOME/weblogic/myserver` or `BEA_HOME/wlserver6.1/config/peoplesoft`

```
Enter WebLogic instance Directory
[/export/home/psft/weblogic/myserver] ?:
```

8. At this prompt, enter the system user ID used to install PeopleSoft software. For example, `psft`

```
Enter PeopleSoft user ID          [psft]      ?:
```

9. At this prompt, enter the primary group of the PeopleSoft user. For example, `sys`

```
Enter PeopleSoft user's primary group      [other]      ?:
```
10. At this prompt, enter the complete path to the directory containing the JRE used by the application server.

```
Enter Directory containing JRE used by PeopleSoft?:
```
11. At this prompt, enter `true` if the Sun JCE provider needs to be installed on the JDK. Else, enter `false`.

```
Install JCE for this ('true' or 'false') [true] ?:
```
12. At this prompt, enter `true` if the Sun JSSE provider needs to be installed on the JDK. Else, enter `false`.

```
Install JSSE for this ('true' or 'false') [true] ?:
```
13. At this prompt, enter the generic PeopleSoft user ID that the web server uses to identify itself to the application server. For example, `DEFAULT_USER`.

```
Enter PeopleSoft DEFAULT_USER [DEFAULT_USER] ?:
```
14. At this prompt, enter the PeopleSoft `DEFAULT_USER` password assigned while creating the `DEFAULT_USER`.

```
Enter PeopleSoft DEFAULT_USER Password?:
```
15. At this prompt, reenter the PeopleSoft `DEFAULT_USER` password.

```
Re-enter Password?:
```
16. At this prompt, enter the name of the PIA site, as given during PeopleSoft installation. For example, `peoplesoft8`.

```
Enter Name of the PIA Site?:
```
17. At this prompt, enter the name of the PIA web application as given during PeopleSoft installation. For example, `PORTAL`.

```
Name of the PIA web application ?:
```

If the information entered is valid, the agent tools will now proceed to configure the instance of the application server. Once this is completed, the tools will display a message indicating the status of the task and the program will exit:

```
Agent Configured Successfully for Application Server Instance.
```

Agent for Apache Tomcat Server 4.1.27

After you have completed the steps outlined in the section [Common Configuration Steps](#), the agent tools will prompt you for information specific to the Tomcat Server instance that is being configured.

1. At the following prompt, enter the location of the configuration directory of the Tomcat Server instance that is being configured:

```
Enter complete path to the configuration directory of the Tomcat Server Instance?:
```

2. At the following prompt, enter `true` if you want to install the agent filter in the global deployment descriptor (`web.xml`). If however, you want to add the filter individually to the application deployment descriptors, enter `false`.

```
Filter installed in global web.xml ('true' or 'false')?:
```

If the information entered is valid, the agent tools will now proceed to configure the instance of the Tomcat Server. Once this operation is completed, the agent tools will display a message indicating the status of the task and the program will exit:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE

Automatic reconfiguration of the web applications `admin` and `manager` to work with the agent is not supported on any Tomcat Server instance other than the default. The default instance is the first Tomcat Server instance configured after the agent was installed.

However, if you have installed the agent filter in your global deployment descriptor (`web.xml`), you may be able to configure these web applications to work with the default instance and the new instance of the agent (that you are configuring using `agentadmin`) just like any other web application.

Please see the following documents for more information on Tomcat server administration and configuration:

- <http://jakarta.apache.org/tomcat/tomcat-4.1-doc>
 - <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/manager-howto.html>
-

Agent for Macromedia JRun 4

After you have completed the steps outlined in the section [Common Configuration Steps](#), the If the information entered is valid, the agent tools configure the Application Server instance. On completion of this task, prior to the program exiting, the tools display the following message indicating status:

```
Agent Configured Successfully for Application Server Instance.
```

tool will prompt you for information specific to the Macromedia JRun 4 instance that is being configured.

1. At the following prompt, enter the location of the configuration directory `SERVER-INF` of the server instance being configured:

```
Enter complete path to the SERVER-INF directory of the JRun Server Instance?:
```

2. At the following prompt, enter `true` if you want to install the agent filter in the global deployment descriptor `default-web.xml`. Or, if you want to add the filter individually to the application deployment descriptors, enter `false`.

```
Install agent filter in default-web.xml ('true' or 'false')?:
```

If the information entered is valid, the `agentadmin` tool will now proceed to configure the instance of Macromedia JRun server. Once this operation is completed, the `agentadmin` tool will display a message indicating the status of the task and the program will exit:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE The agent tool does not automatically configure the administration instance of Macromedia JRun 4. So the administration instance will continue to run with the administrator's user name and password with which it was installed initially. However, you can configure the administration instance manually as explained in the following section.

Configuring the Administration Instance of Macromedia JRun 4

To configure the administration instance of Macromedia JRun 4, do the following:

1. Create the necessary role (`jmadmin`) and user (`admin`) in Sun ONE Identity Server.
2. Assign the user to the role.
3. Define appropriate resources (`http://host:adminport/*`)
4. Define appropriate policies (rules, subjects, etc.) for the resource(s).
5. Configure the agent for administration instance as explained in the previous section.

Agents for Oracle 9iAS R2 and Oracle 10g

If you are configuring the agent for Oracle 9iAS R2 or Oracle 10g, the agent tools will prompt you for information specific to the Oracle 9iAS or Oracle 10g server instance respectively after you have completed the steps outlined in the section [Common Configuration Steps](#).

1. At the following prompt, enter the complete path to the Oracle Server instance configuration directory. Ensure that you have write privileges for this directory.

```
Enter the location of Oracle Instance Config directory?:
```

NOTE If you specify the directory path with a "/" at the end, for example *instance_dir/oracle/config/*, you will have to enter the path in the same form while unconfiguring too.

If the information entered is valid, the agent tools will now proceed to configure the instance of the application server. When this is completed, the tools will display a message indicating the status of the task and the program will exit:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE If the Oracle 9iAS or Oracle 10g instance that has been configured is outside the domain in which the agent was installed, you must complete the post-installation tasks for the respective instance as mentioned in the section ["Post-Installation Tasks"](#) on page 107.

Agent for SAP Enterprise Portal 6.0 SP2

If you are installing the agent for SAP Enterprise Portal 6.0 SP2, after you have completed the steps outlined in the [Common Configuration Steps](#) section, the agent tool will prompt you for information specific to the SAP Enterprise Portal 6.0SP2 Server instance that is to be configured.

1. At the following prompt, enter the complete path to the server instance directory that will be configured:

```
Enter the path to SAP Enterprise Portal Server Directory?:
```

If the information entered is valid, the agent tools will now proceed to configure the instance of the application server. When this is completed, the tools will display a message indicating the status of the task and the program will exit:

Agent Configured Successfully for Application Server Instance.

NOTE

- Before you configure an instance, you must follow the appropriate pre-installation steps as outlined in the section [“Pre-Installation Tasks” on page 31](#).
 - Once the instance has been configured by the agentadmin tool, you must follow the applicable post-installation steps as mentioned in the section [“Post-Installation Tasks” on page 107](#).
-

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

If you are installing the agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, after you complete the steps outlined in [Common Configuration Steps](#), the agent tool displays the following prompt:

Enter the path to SAP Server Directory?:

On the prompt, enter the complete path to the server instance directory that will be configured.

If the information you enter is valid, the agent tools proceed to configure the instance of the application server. When this is completed, the tools display a message indicating the status of the task. The program then exits and this message is displayed:

Agent Configured Successfully for Application Server Instance.

NOTE

- Before you configure an instance, follow the appropriate pre-installation steps as outlined in [“Pre-Installation Tasks” on page 31](#)
 - Once the instance has been configured by the agentadmin tool, follow the applicable post-installation steps described in [“Post-Installation Tasks” on page 107](#)
-

Agent for Sun Java System Application Server 8.1

After you have completed the steps outlined in [“Common Configuration Steps” on page 277](#), the agent tool will prompt you for information specific to the Sun Java System Application Server 8.1 instance that is being configured. Follow these steps:

1. At the following prompt, enter the location of the Domain Config Directory for the Sun Java Application Server 8.1 instance that is being configured:

Enter the Domain config Directory
 [*ApplicationServer-base*/domains/domain1/config]?:

2. At the following prompt, enter the application server instance that is being configured

```
Enter Sun Java(TM) System Application Server Instance Name?:  
[ExampleServer]?
```

3. At the following prompt, enter the port number where Sun Java System Administration Server is available:

```
Enter Sun Java(TM) System Application Server Administrative Server port  
[4849] ?:
```

where 4849 is the default port number.

4. At the following prompt, enter the user name for the Administrator of Sun Java System Application Server 8.1.

```
Enter Sun Java(TM) System Application Server Administrative User Name?:  
admin
```

where admin is the default administrative user name. Of course the actual administrative user name could be different.

5. At the following prompt, enter the password for the Administrator of Sun Java System Application Server 8.1:

```
Enter Sun Java(TM) System Application Server Administrative User  
Password?:
```

6. At the following prompt, re-enter the password for the Administrator of Sun Java System Application Server 8.1:

```
Reenter Sun Java(TM) System Application Server Administrative User  
Password?
```

If the information entered is valid, the agent tools configure the Application Server instance. On completion of this task, prior to the program exiting, the tools display the following message indicating status:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE	<p>For instances where DAS is not on the agent host, therefore DAS is remote, the following applies:</p> <p>The <code>agentadmin</code> tool cannot configure a remote instance. If the server instance is on a different host than the DAS host, then the agent has to be installed on every host that has a server instance on it.</p> <p>For instances belonging to a different domain, the following applies:</p> <p>The <code>Agentadmin</code> tool cannot configure instances within a different domain on the same host. Agents do not support multiple domains per host.</p>
-------------	---

Unconfiguring the Agent for an Application Server Instance

This section outlines the steps necessary to unconfigure an agent for an application server instance.

NOTE	<p>The agent tools for IBM WebSphere Application Server 5.0/5.1 agent do not support the unconfigure option. In order to manually unconfigure the agent for IBM WebSphere Application Server 5.1, you can refer to the steps described in the section Agent for IBM WebSphere Application Server 5.0/5.1 Instance.</p> <p>When the agent is removed from any application server instance, you must ensure that any applications deployed on that particular instance are changed to remove the agent filter component and are restored to their previous state as described in “Pre-Uninstallation Tasks” on page 301.</p>
-------------	--

There are no common steps for unconfiguring all the agents using the agent tools. Hence, you can directly jump to the relevant agent section from the following list to unconfigure the agent for the respective application server instance.

- [Agent for Sun ONE Application Server 7.0](#)
- [Agents for BEA WebLogic Server \(All Versions\)](#)
- [Agent for IBM WebSphere Application Server 5.0/5.1 Instance](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agent for Apache Tomcat Server 4.1.27](#)
- [Agent for Macromedia JRun 4](#)

- [Agent for Oracle 9iAS R2 and Oracle 10g Server Instance](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

1. In a shell prompt from the `bin` directory of the agent installation, invoke the agent tool using the control script as follows:

```
#./agentadmin -unconfigure
```

On Windows platform, use this command:

```
agent_install_dir\IdentityServer\j2ee_agents\bin> agentadmin
-unconfigure
```

2. At the following prompt, enter the location of the config directory for the application server instance that is being unconfigured:

```
Enter the Sun ONE Application Server instance config Directory?:
```

3. At the following prompt, enter the port number where Sun ONE Application Server Administration Server is available. Make sure that the administration server is running when doing this process:

```
Enter Sun ONE Application Server Administrative Server port
[4848]?:
```

4. At the following prompt, enter the user name for the Administrator of Sun ONE Application Server.

```
Enter Sun ONE Application Server Administrative User Name
[admin]?:
```

5. At the following prompt, enter the password for the Administrator of Sun ONE Application Server:

```
Enter Sun ONE Application Server Administrative User Password?:
```

6. At the following prompt, re-enter the password for the Administrator of Sun ONE Application Server:

```
Reenter Sun ONE Application Server Administrative User Password?:
```

If the information entered is valid, the agent tools will now proceed to unconfigure the instance of the application server. Once this is completed, the tools will display a message indicating the status of the task and the program will exit:

Agent Unconfigured Successfully for Application Server Instance.

Agents for BEA WebLogic Server (All Versions)

NOTE If the BEA WebLogic Server instance being unconfigured is outside the domain in which the agent was installed, complete the pre-uninstallation tasks for BEA WebLogic Server instance as mentioned in the chapter [Chapter 5, “Uninstalling the Agent”](#) on page 301.

This section addresses the configuration of agents for any of the following:

- BEA WebLogic Server 6.1 SP2,
 - BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1
 - BEA WebLogic 8.1 SP2/SP3 Server/Portal
1. In a shell prompt from the `bin` directory of the agent installation, invoke the agent tool using the following command:

```
#./agentadmin -unconfigure
```

On Windows platform, use this command:

```
C:\Sun\IdentityServer\j2ee_agents\bin> agentadmin -unconfigure
```

2. At the following prompt, enter the location of the startup script that is used to launch this instance of WebLogic Server or Portal that is being unconfigured:

```
Enter the location of WebLogic Server Startup Script?:
```

If the information entered is valid, the agent tools proceed to unconfigure the instance of the application server. Once this is completed, the tools display a message indicating the status of the task. The program then exits and this message is displayed:

```
Agent Unconfigured Successfully for Application Server Instance.
```

Agent for IBM WebSphere Application Server 5.0/5.1 Instance

The agent tools do not support the unconfiguration option for IBM WebSphere Application Server 5.0/5.1 instances. However, the following list of steps provide the necessary details that will allow you to manually unconfigure an agent for an instance of the application server:

1. Stop the WebSphere Application Server instance that is being unconfigured.
2. Using the Administrative Console, remove the agent-specific classpath and boot classpath entries.

3. Using the Administrative Console, remove the agent-specific JVM options from Generic JVM Arguments.
4. Restart the WebSphere Application Server instance.

NOTE Since WebSphere Application Server relies on a global security scheme, the application server instance that has been unconfigured will continue to operate with the agent realm, which is installed as a part of the earlier agent installation. Once the agent is unconfigured from the instance, the instance will no longer be able to load the agent realm and may fail to start. In order to completely unconfigure this instance, you can either uninstall the agent from this system, or operate the agent for the previous instance in a mode that does not require agent realm and disable the agent realm. Refer to the section on [Agent Filter Modes](#) for information on which agent operation modes do not require agent realm, and the section on [Disabling the Agent Realm](#) for information on how the agent realm can be disabled for IBM WebSphere Application Server instance.

Agent for PeopleSoft 8.3/8.4/8.8

1. In a shell prompt from the `bin` directory of the agent installation, invoke the agent tool using the control script as follows:

```
#!/agentadmin -unconfigure
```

On HP-UX platform, use this command:

```
agent_install_dir\IdentityServer\j2ee_agents\bin> agentadmin
-unconfigure
```

2. At the following prompt, enter the version of the PeopleSoft Application version you want to unconfigure.

```
Enter the PeopleSoft application version ('8.3' or '8.4' or
'8.8') [8.3] ?:
```

3. At this prompt, enter `true` if you have the PeopleSoft application server installed locally. Else, enter `false`.

```
PeopleSoft Application Server installed locally ('true' or
'false') [true] ?:
```

4. At this prompt, enter `true` if you have BEA WebLogic Server installed locally. Else, enter `false`.

```
WebLogic Server installed locally ('true' or 'false') [true] ?:
```

- At this prompt, enter the full path to the directory where the PeopleSoft Application Server is installed. For example, `PS_HOME/appserv`

```
Enter PeopleSoft AppServer Directory
[/export/home/psft/appserv] ?:
```

- At this prompt, enter the PeopleSoft domain that this agent protects. For example, `HDMO`.

```
Enter PeopleSoft Domain Name [HDMO] ?:
```

- At this prompt, enter the full path to the WebLogic instance directory. For example, `PS_HOME/weblogic/myserver`

```
Enter WebLogic instance Directory
[/export/home/psft/weblogic/myserver] ?:
```

If the information entered is valid, the agent tools will now proceed to unconfigure the instance of the application server. Once this is completed, the tools will display a message indicating the status of the task and the program will exit:

```
Agent Unconfigured Successfully for Application Server Instance.
```

Agent for Apache Tomcat Server 4.1.27

To unconfigure the Tomcat Server instance, do the following:

- Stop the Tomcat Server instance you want to unconfigure.
- At a shell prompt from the `bin` directory of the agent installation, invoke the agent tools using the following command:

```
#!/agentadmin -unconfigure
```

On the Windows platform, use this command:

```
agent_install_dir\IdentityServer\j2ee_agents\bin\agentadmin -unconfigure
```

- At the following prompt, enter the path to the configuration directory of the Tomcat Server instance that is being configured:

```
Enter complete path to the configuration directory of the Tomcat Server
Instance?:
```

- At the following prompt, enter `true` if you have installed the agent filter in the global deployment descriptor (`web.xml`). If however, you have added the filter individually to the application deployment descriptors, enter `false`.

```
Filter installed in global web.xml ('true' or 'false')?:
```

NOTE The agent tools will not unconfigure the individual web applications' deployment descriptors, if you had not installed the agent filter globally. Hence, you will need to remove the agent filter manually from the deployment descriptor of every deployed web application.

If the information entered is valid, the agent tools will now proceed to unconfigure the instance of the Tomcat Server. Once this operation is completed, the agent tools will display a message indicating the status of the task and the program will exit:

```
Agent Unconfigured Successfully for Application Server Instance.
```

Agent for Macromedia JRun 4

To unconfigure an instance of Macromedia JRun 4 using agent tools, you must first stop the instance and then start the configuration as explained in the following steps:

1. At the Solaris command prompt, invoke the `agentadmin control` script as follows:

```
# Agent_Install_Dir/SUNWam/j2ee_agents/bin/agentadmin -unconfigure
```

2. At the following prompt, enter the location of the configuration directory (SERVER-INF) of Macromedia JRun server instance being configured:

```
Enter the complete path to the SERVER-INF directory of the JRun 4
Server Instance?:
```

3. At the following prompt, enter `true` if the agent filter was installed in the global deployment descriptor `default-web.xml`. In this case, the unconfiguration program will remove the agent filter. If the filter was added individually to the application deployment descriptors, enter `false`. In this case, `default-web.xml` will remain untouched by the unconfiguration program.

```
Install agent filter in default-web.xml ('true' or 'false')?:
```

If the information entered is valid, the `agentadmin` tool will now proceed to unconfigure the instance of the Macromedia JRun Server 4.0. Once this operation is completed, the `agentadmin` tool will display a message indicating the status of the task and the program will exit:

```
Agent Unconfigured Successfully for Application Server Instance.
```

CAUTION

When you uninstall the agent or unconfigure a Macromedia JRun 4 server instance, the `jrun_instance_name_jvm.config` file is simply removed. You must make sure to back up or save any custom configuration data that might have been added to this file.

Agent for Oracle 9iAS R2 and Oracle 10g Server Instance

Use the following steps to unconfigure the application server instance of Oracle 9iAS or Oracle 10g:

1. In a shell prompt from the `bin` directory of the agent installation, invoke the agent tool using the following command:

```
# ./agentadmin -unconfigure
```

2. At the following prompt, enter the complete path to the Oracle Server instance configuration directory. Ensure that you have write privileges for this directory.

```
Enter the location of Oracle Instance Config directory?:
```

If the information entered is valid, the agent tools will now proceed to unconfigure the instance of the application server.

NOTE While configuring the instance directory, if you had specified the directory path with a "/" at the end, for example `instance_dir/oracle/config/`, you must enter the path in the same form while unconfiguring too.

Once this is completed, the tools will display a message indicating the status of the task and exit:

```
Agent Unconfigured Successfully for Application Server Instance.
```

Agent for SAP Enterprise Portal 6.0 SP2

Perform the following steps to unconfigure the SAP Enterprise Portal server instance:

1. In a shell prompt from the `bin` directory of the agent installation, invoke the agent tools using the following command:

```
# ./agentadmin -unconfigure
```

2. At the following prompt, enter the location of the startup script that is used to launch this instance of WebLogic Server which is being unconfigured:

Enter the path to SAP Enterprise Portal Server Directory?:

If the information entered is valid, the agent tools will now proceed to unconfigure the instance of the application server. Once this is completed, the tools will display a message indicating the status of the task and the program will exit:

```
Agent Unconfigured Successfully for Application Server Instance.
```

NOTE

- Before you unconfigure an instance, you must follow the appropriate pre-uninstallation steps as outlined in [“Pre-Uninstallation Tasks” on page 301](#).
 - Once the instance has been unconfigured by the agentadmin tool, you must follow the applicable post-uninstallation steps as mentioned in [“Post-Uninstallation Tasks” on page 324](#).
-

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

Perform the following steps to unconfigure the SAP server instance:

1. In a shell prompt from the bin directory of the agent installation, invoke the agent tools using the following command:

```
./agentadmin -unconfigure
```

The following prompt appears:

```
Enter the path to SAP Server Directory?:
```

2. Enter the location of the server directory which is being unconfigured.

If the information you enter is valid, the agent tools proceed in unconfiguring the instance of the application server. Once the process is complete, the tools display a message indicating the status of the task. The program then exits and this message is displayed:

```
Agent Unconfigured Successfully for Application Server Instance.
```

NOTE

- Before you unconfigure an instance, follow the appropriate pre-uninstallation steps as outlined in [“Pre-Uninstallation Tasks” on page 301](#)
 - Once the instance has been configured by the agentadmin tool, follow the applicable post-uninstallation steps described in [“Post-Uninstallation Tasks” on page 324](#)
-

Agent for Sun Java System Application Server 8.1

Perform the following steps to unconfigure the Sun Java System Application Server 8.1 instance:

Enter Sun Java(TM) System Application Server Administrative User Name?:
admin

Enter Sun Java(TM) System Application Server Administrative User Password?:

ReEnter Sun Java(TM) System Application Server Administrative User Password?:

1. At the following prompt, enter the location of the Domain Config Directory for the Sun Java Application Server 8.1 instance that is being unconfigured:

Enter the Domain config Directory
[*ApplicationServer-base*/domains/domain1/config]?:

2. At the following prompt, enter the application server instance that is being configured

Enter Sun Java(TM) System Application Server Instance Name?:
[ExampleServer]?

3. At the following prompt, enter the port number where Sun Java System Administration Server is available:

Enter Sun Java(TM) System Application Server Administrative Server port
[4849] ?:

where 4849 is the default port number.

4. At the following prompt, enter the protocol used by Application Server. This protocol value may either be HTTP or HTTPS:

Enter Sun Java(TM) System Application Server Administrative Server
protocol [https] ?:

5. At the following prompt, enter the user name for the Administrator of Sun Java System Application Server 8.1.

Enter Sun Java(TM) System Application Server Administrative User Name?:
admin_user_name

where admin is the default administrative user name. Of course the actual administrative user name could be different.

6. At the following prompt, enter the password for the Administrator of Sun Java System Application Server 8.1:

```
Enter Sun Java(TM) System Application Server Administrative User
Password? :
```

7. At the following prompt, re-enter the password for the Administrator of Sun Java System Application Server 8.1:

```
Reenter Sun Java(TM) System Application Server Administrative User
Password?
```

If the information entered is valid, the agent tools configure the Application Server instance. On completion of this task, prior to the program exiting, the tools display the following message indicating status:

```
Agent Configured Successfully for Application Server Instance.
```

NOTE For instances where DAS is not on the agent host, therefore DAS is remote, the following applies: the `agentadmin` tool cannot unconfigure a remote instance.

Agent APIs

The agent runtime provides access to all the Sun ONE Identity Server APIs that can be used to further enhance the security of your application. On top of these APIs, the agent also provides a set of APIs that allow the application to find out the SSO Token string associated with the logged on user. These APIs can be used from within the Web Container or the EJB Container of the application server.

The following sections illustrate the available agent APIs that can be used from within an application:

Class AmFilterManager

```
com.sun.identity.agents.filter.AmFilterManager
```

Available API

- `public static com.sun.identity.agents.filter.AmSSOCache
getAmSSOCacheInstance() throws
com.sun.identity.agents.arch.AgentException`

This method returns an instance of [Class AmSSOCache](#), which can be used to retrieve the SSO Token for the logged on user. This method can throw an `AgentException` if an error occurs while processing this request.

Class AmSSOCache

`com.sun.identity.agents.filter.AmSSOCache`

Available API

- `public java.lang.String
getSSOTokenForUser(javax.servlet.http.HttpServletRequest request)`

This method returns the SSO Token for the logged on user whose request is currently being processed in the application server's Web Container. This method can return null if the requested token is not available at the time of this call.

- `public java.lang.String getSSOTokenForUser(javax.ejb.EJBContext
context)`

This method returns the SSO Token for the logged on user whose request is currently being processed in the application server's EJB Container. This method can return null if the requested token is not available at the time of this call.

NOTE The API `getSSOTokenForUser(javax.ejb.EJBContext)` can be used only when the agent operation mode is either `J2EE_POLICY` or `ALL`.

Uninstalling the Agent

When you install the J2EE Policy Agent, an uninstallation program is created in the installation directory. Using this uninstallation program, you can remove the agent from your system, if required. While the uninstallation program deletes all the installed files from your system, certain files such as resource files and audit log messages are not deleted.

This chapter describes in detail how to use the agent uninstallation program. Topics covered in this chapter include:

- [Pre-Uninstallation Tasks](#)
- [Launching the Uninstallation Program](#)
- [Using the Uninstallation Program](#)
- [Post-Uninstallation Tasks](#)

Pre-Uninstallation Tasks

The following tasks must be performed before you begin uninstalling the agent from your system.

1. Undeploy any protected application(s) from your application server. Refer to the documentation provided with your application server for details on how this can be done.
2. Restore the deployment descriptors of these applications by their original deployment descriptors.

Having done the above tasks, you must follow the pre-uninstallation tasks specific to your agent covered in these sections. Subsequently, you can move to the section [Launching the Uninstallation Program](#) for steps to uninstall the agent.

- [Agent for Sun ONE Application Server 7.0](#)
- [Agent for BEA WebLogic Server 6.1 SP2](#)
- [Agent for BEA WebLogic Server 7.0 SP2/8.1](#)
- [Agent for IBM WebSphere 5.0/5.1](#)
- [Agent for PeopleSoft 8.3/8.4/8.8](#)
- [Agents for Oracle 9iAS R2 and Oracle 10g](#)
- [Agent for Tomcat Server 4.1.27](#)
- [Agent for SAP Enterprise Portal 6.0 SP2](#)
- [Agent for Macromedia JRun 4](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)
- [Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal](#)
- [Agent for Sun Java System Application Server 8.1](#)

Agent for Sun ONE Application Server 7.0

Before you begin the uninstallation of the agent for Sun ONE Application Server 7.0, you must make sure that:

- The administration server for the instance of application server where the agent is installed is running
- The application server instance being protected by the agent is shut down.

Agent for BEA WebLogic Server 6.1 SP2

Before you begin uninstalling the agent for BEA WebLogic Server 6.1 SP2, you must first manually remove the agent realm that was configured after the installation of this agent. This can be done using the following steps:

NOTE This section outlines the steps necessary to successfully remove the agent realm from the WebLogic Server. It must be noted that the information provided in this section is only to facilitate the removal of agent realm and should not be taken as a substitute for the information provided in WebLogic Server documents. For a complete in-depth discussion on WebLogic Custom Realms, refer WebLogic Server documentation at: <http://www.bea.com>.

Selecting a Different Caching Realm

1. Log on to the BEA WebLogic Administration console.
2. Click on the Security node on the left hand menu. The console displays the current security configuration on the right side content pane.
3. In the right content pane, click on the Filerealm tab. In the form that is now displayed, select any caching realm other than Agent Caching Realm from the drop down menu labeled Caching Realm.
4. Click on the Apply button and restart the Application Server.

Removing the Agent Caching Realm

1. Log on to the Administration Console.
2. Navigate to the node Security > Caching Realms on the left hand menu. This displays a list of configured caching realms on the right side.
3. From the list of displayed caching realms on the right side, click on the delete icon next to the Agent Caching Realm. When prompted to confirm your action, click the Yes button.
4. Restart the Application Server.

Removing the Agent Realm

1. Log on to the BEA WebLogic Administration console.
2. Navigate to the node Security > Realms on the left hand menu. This displays a list of configured realms on the right side.
3. From the list of displayed realms on the right side, click on the delete icon next to the agent realm. When prompted to confirm your action, click the Yes button.
4. Restart the Application Server.

The agent realm is now removed from the WebLogic Server configuration.

5. Once the above steps are complete, you must shutdown the BEA WebLogic Server before launching the agent uninstallation program.

Agent for BEA WebLogic Server 7.0 SP2/8.1

Before you begin uninstalling the agent for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1, you must first manually remove the Agent Authentication Provider that was configured after the installation of this agent. This can be done using the following steps:

NOTE This section outlines the steps necessary to successfully remove the Agent Authentication Provider from the WebLogic Server. It must be noted that the information provided in this section is only to facilitate the removal of Agent Authentication Provider and should not be taken as a substitute for the information provided in WebLogic Server documents. For a complete in-depth discussion on WebLogic Custom Realms, refer WebLogic Server documentation at: <http://www.bea.com>.

1. Logon to the WebLogic Server console.
2. On the left frame, click on *agentdomain* > Security > Realms > myrealm, where *agentdomain* is the domain you had configured.
3. On the right frame, click on the Providers tab.
4. Click on Authentication.
5. Look for Agent Authenticator and click the Delete icon next to it.
6. Confirm the delete by clicking Yes.
7. If the DefaultAuthenticator is the only other authentication provider, change its control flag to REQUIRED. Click Apply.
8. Restart the server. The Agent Authentication Provider is now removed from the WebLogic Server configuration.

Once these steps are complete, you must shutdown the BEA WebLogic Server before launching the agent uninstallation program.

Agent for IBM WebSphere 5.0/5.1

Before you launch the uninstallation program for the agent for IBM WebSphere 5.0/5.1 Application Server, you must ensure that the WebSphere Application Server instance is running and the Administration application has been deployed on this server.

Agent for PeopleSoft 8.3/8.4/8.8

Before you launch the uninstallation program for the agent for PeopleSoft 8.3/8.4/8.8, you must make sure that all PeopleSoft processes have been shutdown including any front-end web server processes as applicable to your configuration.

Agents for Oracle 9iAS R2 and Oracle 10g

Stop all processes related to Oracle 9iAS R2 or Oracle 10g before uninstalling the agent. To learn more about stopping these processes, refer to the Administration Guides at the following URLs:

- Oracle 9iAS R2

http://download-west.oracle.com/docs/cd/A97329_03/core.902/a92171/toc.htm

- Oracle Application Server 10g

http://download-west.oracle.com/docs/cd/B10464_03/core.904/b10376/toc.htm

Not stopping all processes related to Oracle such as `emctl`, `opmnctl`, `dcmctl` can lead to the malfunctioning of the agent resulting in the application or its portions becoming inaccessible.

Agent for Tomcat Server 4.1.27

Before you launch the uninstallation program for the agent for Tomcat 4.1.27 Server, you must ensure that the Tomcat server has been shutdown completely.

Agent for SAP Enterprise Portal 6.0 SP2

Before you launch the uninstallation program for the agent for SAP Enterprise Portal 6.0 SP2, you must first shut down the SAP Enterprise Portal completely.

Agent for Macromedia JRun 4

Before you launch the uninstallation program for the agent for Macromedia JRun 4, you must ensure that JRun server has been shutdown completely.

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

Before you launch the uninstallation program for the agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1, you must shut down the SAP server completely. If any agent protected applications are deployed on the Web Application server, remove them from the server before you shut the server down. In order to restore these applications to their original state, perform the restoration procedures as follows:

- [Restore Deployment Descriptors](#)
- [Restore Library References](#)

Restore Deployment Descriptors

This step is not applicable if you do not have any agent protected applications deployed on the SAP Web Application Server.

In order to fully restore your applications to their original state before agent installation, you must undo the changes done in the post-installation steps outlined in [“Post-Installation Tasks” on page 107](#). Perform the following:

1. Remove the agent filter.

Edit the `web.xml` deployment descriptor of your application to remove the agent filter element.

2. Remove agent specific log in configuration.

If any changes were made to enable web-tier declarative security in the `web.xml` deployment descriptor for your application during agent installation, manually change the deployment descriptor back to its original state in order to fully restore the application.

Restore Library References

During the agent installation, a reference to the `AmSAPAgent` library was added for every protected application in the `library.txt` file located in the following directory:

base-instancedir/cluster/server/managers

In order to restore the system to its original state, edit this file and remove all the references to the `AmSAPAgent` library that were manually added during the agent installation.

Agent for BEA WebLogic 8.1 SP2/SP3 Server/Portal

Before you uninstall the agent for BEA WebLogic 8.1 Server/Portal, manually remove the agent authentication provider that was configured after the installation of the agent. Remove the provider by performing the following steps:

NOTE This section outlines the steps necessary to successfully remove the agent authentication provider from the WebLogic 8.1 Server/Portal. Note that the information provided in this section is only to facilitate the removal of the agent authentication provider and should not be taken as a substitute for the information provided in WebLogic 8.1 Server/Portal documents. For a complete in-depth discussion on WebLogic Custom Realms, refer to WebLogic Server documentation at: <http://www.bea.com>

1. Log on to the WebLogic 8.1 Server console.
2. In the left frame, click *agentdomain*> Security> Realms> myrealm
Where *agentdomain* is the domain you had configured.
3. In the right frame, click the Providers tab.
4. Click Authentication.
5. Click the Delete icon next to Agent Authenticator.
6. Click Yes to confirm the deletion.
7. If Default Authenticator is the only other authentication provider, change its control flag to `REQUIRED`.
8. Click Apply.
9. Restart the server.

The agent authentication provider has now been removed from the WebLogic 8.1 Server/Portal configuration.

Shut down the WebLogic 8.1 Server/Portal domain before launching the agent uninstallation program.

Agent for Sun Java System Application Server 8.1

Before you uninstall the policy agent for Sun Java System Application Server 8.1, ensure that the following conditions apply.

- The following server is running:
 - Domain Administration Server (DAS) for the application server instance that the agent is protecting
- Preferably, the following server has been shut down (this is a standard precautionary measure):
 - Application Server instance that will be protected by the agent

Launching the Uninstallation Program

After completing the pre-uninstallation tasks, you are now ready to launch the uninstallation program. This uninstallation program is platform-specific and should be used in accordance with the steps outlined in this section. Once the uninstallation program has been launched successfully, you may jump to the next section which provides the necessary details on how to use this uninstallation program.

Launching the Uninstallation Program on Solaris, HP-UX, AIX and Linux

The following steps provide details on how the agent uninstallation program can be launched on Solaris, HP-UX, AIX and Red Hat Advanced Server 2.1 platforms.

1. Login as the root user and go to the directory where the agent is installed.

2. Set your `JAVA_HOME` environment variable to JDK version 1.3.1 or higher. If your system does not have the required version of JDK, you may either download and install a compatible version of this software from the Java web site <http://java.sun.com> or use the JDK provided along with the application server.

The uninstallation program provides two types of interfaces—a GUI or a graphical user interface, and an interactive command line interface. In most cases, the GUI uninstallation program can be used for installing the agent. However, in cases such as when you are uninstalling the agent over a telnet session on a remote server and do not have windowing capabilities, then it is recommended that you use the command-line uninstallation program for uninstalling the agent.

If you choose to use the GUI uninstallation program, then it is required that you set your `DISPLAY` environment variable to ensure that the GUI uninstallation program window appears on the correct console.

3. Once you have completed the above steps, you are now ready to execute the uninstallation script that launches the agent uninstallation program. This program can be launched in two different modes—a GUI mode which has a rich graphical interface, and an interactive terminal mode that does not require any windowing capability. The following steps may be used to launch the uninstallation program in each of these two modes.

NOTE In order to launch the uninstallation program on Red Hat Advanced Server 2.1, you must have the Korn shell installed in the system. If the Korn shell is not available, you may launch the uninstallation program directly by using a Java command such as `java -cp . classfilename`

Launching the Uninstallation Program in the GUI Mode

1. To launch the uninstallation program in the GUI mode, use the following command:

```
#./uninstall
```

When launching the program in a GUI mode, it is required that you set your `JAVA_HOME` and `DISPLAY` environment variables correctly as pointed out in the previous steps. If however, you have not set these variables, the setup script will prompt you for their values accordingly. If your `JAVA_HOME` environment variable is not set correctly, the setup script will display the following prompt:

```
Enter JAVA_HOME location (Enter "." to abort):
```

2. At this prompt, you may type the full path to the JDK installation directory that should be used for launching the uninstallation program. Otherwise, enter a period (.) to abort the uninstallation.

If your `DISPLAY` environment variable is not set correctly, the setup script will display the following prompt:

Please enter the value of `DISPLAY` variable (Enter "." to abort):

3. At this prompt, you may specify the hostname for the `DISPLAY` variable. Otherwise enter a period (.) to abort the uninstallation.

NOTE

- If you enter a value for the `DISPLAY` environment variable which is not appropriate, the uninstallation program window may get displayed on a different console, giving the impression that the uninstallation program is hanging. If such a condition occurs, please verify your `DISPLAY` environment variable by launching another graphical program such as `xterm`.
 - If the supplied value for `DISPLAY` environment variable is not a valid or reachable hostname, or is that of a host, which does not allow you to use its windowing capabilities, the uninstallation program will automatically shift to the interactive command line mode.
-

Launching the Uninstallation Program in the Command-Line Mode

1. To launch the uninstallation program in a non-GUI or command line mode, use the following command:

```
# ./uninstall -nodisplay
```

When launching the program in a command line mode, it is required that you set your `JAVA_HOME` environment variable correctly as pointed out in the previous steps. If however, you have not set this variable, the uninstallation script will prompt you for its value as follows:

Enter `JAVA_HOME` location (Enter "." to abort):

2. At this prompt, you may type the full path to the JDK installation directory that should be used for launching the uninstallation program. Otherwise, enter a period (.) to abort the uninstallation.

Depending upon the mode in which you have launched the uninstallation program, you should see the appropriate interface appear at this stage.

NOTE

Instead of using the provided script to launch the uninstallation program, you may alternatively invoke the uninstaller class file in the agent installation directory with a JDK runtime version 1.3.1 or above to launch the uninstallation program.

Launching the Uninstallation Program on Windows 2000

Follow these steps to launch the agent uninstallation program on Windows 2000 platform:

1. Log into your Windows system as a user with Administrative privileges. If you do not have administrative privileges, either log on as Administrator user or request such privileges to be granted to your account by the system administrator of the machine or domain as applicable.
2. Open a command prompt, and go to the directory where the agent is installed. This directory will contain the `uninstall.bat` file, which can be used to launch the uninstaller.
3. In order to use `uninstall.bat` to launch the uninstallation program, you must have a JDK version 1.3.1 or higher available in your system path. This can be verified by typing the following command in a command prompt window:

```
C:\> java -version  
  
java version 1.3.1_02  
  
Java(TM) 2 Runtime Environment, Standard Edition (build  
1.3.1_02-b02)  
  
Java HotSpot(TM) Client VM (build 1.3.1_02-b02, mixed mode)
```

NOTE It is recommended that you download and install a compatible version of JDK if you do not have one on your system. Using the JDK supplied by the application server may result in malfunction of the uninstallation program and you may not be able to successfully uninstall the agent.

The program `uninstall.bat` may be executed by typing the file name at the command prompt window in a directory where it is present, or by double clicking the file in Windows Explorer. For example:

```
C:\> uninstall.bat
```

The uninstallation program provides two types of interfaces—a GUI and an interactive command line interface. You can launch the uninstallation program in the GUI mode by invoking `uninstall.bat` file from a command prompt window as shown above or by double clicking it in Windows Explorer. The uninstallation program may be launched in a command line mode by passing the argument `-nodisplay` to the `uninstall.bat` script as follows:

```
C:\> uninstall.bat -nodisplay
```

NOTE Instead of using the provided scripts and executable that launch the uninstallation program, you may alternatively invoke the uninstaller class file in the agent installation directory, with a JDK runtime version 1.3.1 or above to launch the uninstallation program.

Depending upon the mode in which you have launched the uninstallation program, you should see the appropriate interface appear at this stage.

Using the Uninstallation Program

As mentioned in the last section, the agent uninstallation program provides two types of interfaces - a GUI or graphical user interface, and a non-GUI or console based interactive interface. Once the uninstallation program has been launched, you must provide all the necessary information requested by this program in order to successfully uninstall the agent on your system. The following two sections describe in detail how to use this uninstallation program in each interface to successfully uninstall the agent on your system.

Once the uninstallation program has been launched, the expected interaction will be the same on any platform for a given application server agent. Thus the steps outlined here for a certain application server agent uninstallation will be applicable to the same agent uninstallation for the same application server on a different platform.

Using the GUI Uninstallation Program

When the agent uninstallation program is launched in the GUI mode, it presents the user with a series of screens that gather the necessary information and report the status of the uninstallation progress. The agent uninstallation also provides active feedback, by means of pop-up dialog boxes that contain the necessary help or error messages, to help the user enter correct information in case the provided information is not valid. Follow these steps to use the agent uninstallation program in the GUI mode.

NOTE The information regarding the use of the agent uninstallation program as outlined in this section is applicable to all application server agents supported in this release. Any agent-specific differences have been identified here separately.

1. Launch the uninstallation program as explained in the earlier sections. The uninstallation program begins with a welcome screen.
2. Read the information provided in this screen.
3. Click Next to continue. The screen that follows is specific to the application server agent that is being uninstalled. Please jump to the appropriate section from the following list depending on which application server agent is being uninstalled.

[Agent for Sun ONE Application Server 7.0](#)

[Agent for IBM WebSphere Application Server 5.0/5.1](#)

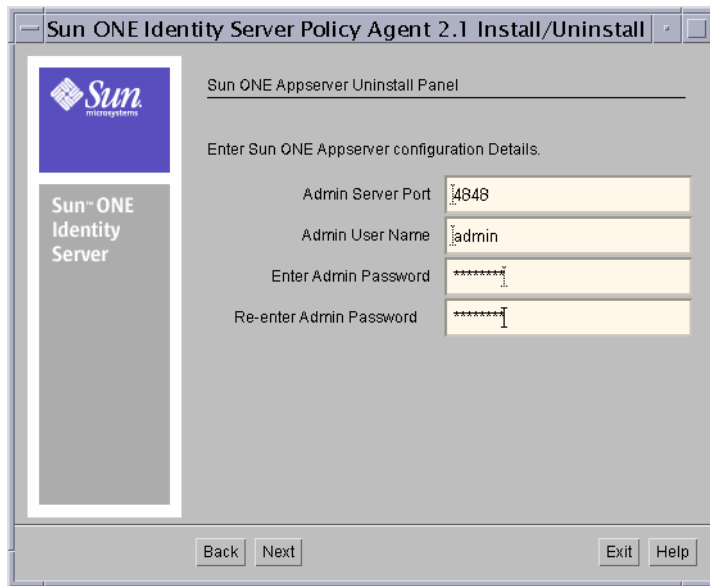
[Agent for Sun Java Systems Application Server 8.1](#)

NOTE When uninstalling any agent for which the option to install JCE or JSSE libraries was selected during installation, make sure that the uninstallation program uses a JDK other than the one using which these libraries were installed. This is because, if the uninstallation program is using the same JDK, it may not successfully remove the JCE and JSSE libraries.

Agent for Sun ONE Application Server 7.0

In the following screen enter information about the Sun ONE Application Server 7.0 instance, on which the agent is installed.

Figure 5-1 Sun ONE Application Server Uninstallation screen



Admin Server Port: Enter the port number on which the Administration Server for this Sun ONE Application Server 7.0 is available.

Admin User Name: Enter the user name of the Administrator of this Sun ONE Application Server 7.0 instance.

Enter Admin Password: Enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

Re-enter Admin Password: Re-enter the password for the Administrator of this Sun ONE Application Server 7.0 instance.

NOTE You must ensure that the Sun ONE Application Server 7.0 Administration Server is running at the time of uninstallation of the agent. Also ensure that the application server instance on which the agent is installed is shutdown.

Agent for IBM WebSphere Application Server 5.0/5.1

In this screen, enter information about the WebSphere Application Server 5.0/5.1 instance, on which the agent is installed.

Figure 5-2 WebSphere Application Server Uninstallation screen

Sun ONE Identity Server Policy Agent 2.1 Install/Uninstall

Sun Microsystems

Sun ONE Identity Server

WebSphere Application Server Uninstall Panel

Enter WebSphere configuration Details.

WebSphere Admin Connector SOAP RMI

WebSphere Admin Connector Port 8880

amldapuser Password *****

Re-enter amldapuser Password *****

Back Next Exit Help

WebSphere Admin Connector: Specify the protocol for connecting to WebSphere Administration Service. You can choose between SOAP and RMI.

WebSphere Admin Connector Port: Enter the Administration Service Port for specified Connector.

amldapuser Password: Enter the password assigned to the amldapuser as specified during Sun ONE Identity Server installation.

Re-enter amldapuser Password: Re-enter the amldapuser password in this field to ensure that the correct value is used by the agent.

NOTE You must ensure that the IBM WebSphere Application Server 5.0/5.1 instance is running at the time of uninstallation of the agent.

The uninstallation program now checks for the used disk space and displays the Ready to Uninstall screen.

4. Click the Uninstall Now button to start the uninstallation of the agent on your system.

The next screen displays the progress of uninstallation as the uninstallation program makes changes to your system. This screen does not require any user input or action and will automatically proceed to the next screen when the uninstallation program finishes making the necessary changes to your system.

The next screen displays the uninstallation summary. The uninstallation program displays the status of the uninstallation in this screen. You may click on the Details button to see more information on the actions that were performed by the uninstallation program.

The agent has now been removed from your system. You may wish to restart the application server and ensure that it is back to its original state and is fully function.

NOTE The uninstallation program for the IBM WebSphere Application Server 5.0/5.1 agent stops the IBM WebSphere Application Server at the end of the uninstallation. You may manually restart it later.

Agent for Sun Java Systems Application Server 8.1

If you are uninstalling the agent for Sun Java System Application Server 8.1, enter the configuration information on the screen as explained in this section.

Figure 5-3 Sun Java System Application Server 8.1 Uninstallation Screen

Sun Java(TM) System Appserver Uninstall Panel

Enter Sun Java(TM) System Appserver configuration Details.

Admin Server Host

Admin Server Host is Remote?

Admin Server Port

Admin Server Protocol http https

Admin User Name

Enter Admin Password

Re-enter Admin Password

Back Next Exit

Admin Server Host: Enter the host on which DAS for this instance is running.

Admin Server Host is Remote: Enable this field only if DAS for this instance is not on the local host where the instance is running.

Admin Server Port: Enter the port number on which DAS for this instance is available.

Admin Server Protocol: Select the appropriate protocol of DAS. This protocol value may either be HTTP or HTTPS.

Admin User Name: Enter the authorized domain Application Server administrative username.

Enter Admin Password: Enter the Administrator password for the domain. The password is defined as `AS_ADMIN_PASSWORD` in the password file.

Re-enter Admin Password: Re-enter the Administrator password for the domain.

Using the Command Line Uninstallation Program

The uninstallation program consists of one or more message screens that provide the user with information and let the user enter preferences that determine how Sun ONE Identity Server Policy Agent is uninstalled. Some questions require more detailed information that the user may be required to type. The question may have a default value that is displayed inside of brackets []. The default answer can be accepted by, pressing the Enter key. If a different answer needs to be provided, it can be typed at the command prompt.

The agent uninstallation program also provides active feedback to help the user enter correct information in case the provided information is not valid by means of necessary help or error messages. Follow these steps to use the agent uninstallation program in the command line mode.

NOTE The information regarding the use of the agent uninstallation program as outlined in this section is applicable to all application server agents supported in this release. Any agent-specific differences have been identified here separately.

1. Launch the uninstallation program in the `nodisplay` mode, from the agent installation directory. The uninstallation program begins with a welcome message.

2. Read the information provided on this screen and press Enter to continue.

The questions that follow are specific to the application server agent that is being uninstalled. Please jump to the appropriate section from the following list depending on which application server agent is being installed.

[Agent for Sun ONE Application Server 7.0](#)

[Agent for IBM WebSphere Application Server 5.0/5.1](#)

[Agent for Sun Java Systems Application Server 8.1](#)

NOTE

- There is no application server specific question for BEA WebLogic Server and PeopleSoft agent uninstallation programs. If you are uninstalling these agents, please jump to the next step directly.
- When uninstalling the agent for BEA WebLogic Server 6.1 SP2, make sure that the uninstallation program uses a JDK other than the one supplied with BEA WebLogic Server 6.1 SP2. This is because, if the uninstallation program is using the JDK provided with BEA WebLogic Server 6.1 SP2, it does not remove the JCE and JSSE libraries.

Agent for Sun ONE Application Server 7.0

At the prompts, enter information about the Sun ONE Application Server 7.0 instance, on which the agent is installed.

NOTE

You must ensure that the Sun ONE Application Server 7.0 Administration Server is available at the time of uninstallation of the agent. Also ensure that the application server instance on which the agent is installed is shutdown.

```
Admin Server Port [4848] {"<" goes back, "!" exits}  
Admin User Name [admin] {"<" goes back, "!" exits}  
Enter Admin Password [] {"<" goes back, "!" exits}  
Re-enter Admin Password [] {"<" goes back, "!" exits}
```

Admin Server Port: Enter the Admin Port Number for Application Server.

Admin User Name: Enter the Admin Username for Application Server.

Enter Admin Password: Enter the Admin Password for Application Server.

Re-enter Admin Password: Re-enter the Administrator password for Application Server.

Agent for IBM WebSphere Application Server 5.0/5.1

At the prompts, enter information about the IBM WebSphere Application Server 5.0/5.1 instance, on which the agent is installed.

NOTE You must ensure that the IBM WebSphere Application Server 5.0/5.1 instance is running at the time of uninstallation of the agent.

```
WebSphere Admin Connector [SOAP] {"<" goes back, "!" exits}  
WebSphere Admin Connector Port [8880] {"<" goes back, "!" exits}  
Enter amldapuser Password [] {"<" goes back, "!" exits}  
Re-enter amldapuser Password [] {"<" goes back, "!" exits}
```

WebSphere Admin Connector: Specify the protocol for connecting to WebSphere Administration Service. Valid values are SOAP and RMI

WebSphere Admin Connector Port: Enter the Admin Service Port for specified Connector.

amldapuser Password: Enter the amldapuser password.

Re-enter amldapuser Password: Re-enter amldapuser password.

3. In the Ready to Uninstall screen, enter 1 to start the uninstallation of the agent. Enter 2 to restart the uninstallation process from the beginning and enter 3 if you want to exit the uninstallation program.

```
Product Name: Sun ONE Identity Server Policy Agent
Location: C:\Sun
Space Used: 3.85 MB
-----
Sun ONE Identity Server Policy Agent for Sun ONE Application
Server 7.0
Ready to Uninstall
1. UnInstall Now
2. Start Over
3. Exit Uninstallation
What next [1] {"<" goes back, "!" exits}?
```

The next screen displays the progress of uninstallation as the uninstallation program makes changes to your system. This screen does not require any user input or action and will automatically proceed to the next screen when the uninstallation program finishes making the necessary changes to your system.

```
Uninstalling Sun ONE Identity Server Policy Agent
|-1%-----25%-----50%-----75%-----100%|
```

The next screen displays the uninstallation summary as follows. The uninstallation program displays the status of the uninstallation in this screen. You can enter 1 to see more information on the actions that were performed by the uninstallation program.

```

Uninstallation Details:
Product                                Result    More Information
1. Sun ONE Identity Server Policy Agent Full    Available
2. Done
Enter the number corresponding to the desired selection for more
information, or enter 2 to continue [2] {"!" exits}:

```

The agent has now been removed from your system. You may restart the application server and ensure that it is back to its original state and is fully functional.

NOTE The uninstallation program for the IBM WebSphere Application Server 5.0/5.1 agent stops the IBM WebSphere Application Server at the end of the uninstallation. You may manually restart it later.

Agent for Sun Java Systems Application Server 8.1

At the prompts, enter information about the Sun Java System Application Server 8.1 instance, on which the agent is installed as shown in the table and explanations that follow.

```

Admin Server Host [arth] {"<" goes back, "!" exits}
Admin Server Host is Remote [false] {"<" goes back, "!" exits}?
Admin Server Port [4849] {"<" goes back, "!" exits}
Admin Server Protocol [https] {"<" goes back, "!" exits}
Admin User Name [admin] {"<" goes back, "!" exits}
Enter Admin Password [] {"<" goes back, "!" exits}
Re-enter Admin Password [] {"<" goes back, "!" exits}

```

Admin Server Host: Enter the host on which the Domain Administration Server for this instance is running.

Admin Server Host is Remote: Enable this field only if the Domain Administration Server for this instance is not on the local host where the instance is running.

Admin Server Port: Enter the port number on which the Domain Administration Server for this instance is available.

Admin Server Protocol: Select the appropriate protocol of the Domain Administration Server. This protocol value may either be HTTP or HTTPS.

Admin User Name: Enter the authorized domain application server administrative username

Enter Admin Password: Enter the administrator password for the domain. The password is defined as AS_ADMIN_PASSWORD in the password file.

Re-enter Admin Password: Re-enter the Administrator password for the domain.

NOTE For instances where DAS is not on the agent host, therefore DAS is remote, undo the post-installation tasks that were completed after installation as described in the *Post-Installation Tasks* section of Chapter 2, specifically in “[Agent for Sun Java System Application Server 8.1.](#)”

Post-Uninstallation Tasks

The following agents require certain post-uninstallation tasks to be performed before the system is restored to its original state:

- [Agent for SAP Enterprise Portal 6.0SP2](#)
- [Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1](#)

Agent for SAP Enterprise Portal 6.0SP2

When the agent for SAP Enterprise Portal is uninstalled, it modifies various files in the SAP Enterprise Portal installation and takes backups where necessary. When the agent uninstallation program requires multiple changes to a particular file, it may result in more than one backup being made depending upon the uninstallation requirements.

These generated backup files can be easily identified by their suffix of the form `-Agent-timestamp`. For example if a file named `sample.xml` is being modified by the uninstallation program on `03/24/04`, it may create a backup such as `sample.xml-Agent-20040325`.

It is recommended that these sample files be removed from their respective directories and copied to another location outside of the server directory before the SAP Enterprise Portal is started for the first time after the uninstallation of the agent. This is necessary to ensure that no backup file is mistaken for a configuration file by the SAP Enterprise Portal runtime. Moreover, even though the files left behind are harmless, they can cause clutter in the long run and can be difficult to remove if the SAP Enterprise Portal has marked them for being stored in its database.

These files are located in the following directories. The variable `base_instance_dir` referred here denotes the directory where the server instance is located.

- `base_instance_dir`
- `base_instance_dir/ume`
- `base_instance_dir/cluster/server`
- `base_instance_dir/cluster/server/managers`
- `base_instance_dir/cluster/server/services/servlet_jsp/work/jspTemp/irj/root/WEB-INF`

NOTE Do not delete these backup files. If the agent uninstallation gets corrupted, these files are necessary for a manual restoration.

Agent for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1

When the agent for SAP Enterprise Portal and Web Application Server is uninstalled, it modifies various files in the SAP Enterprise Portal installation and/or Web Application Server installation and takes backups where necessary. When the agent uninstallation program requires multiple changes to a particular file, the result might be more than one backup being made depending upon the uninstallation requirements.

These generated backup files can be easily identified by their suffix of the form:
-Agent-*timestamp*

For example if a file named `sample.xml` is modified by the uninstallation program on 11/24/04, it might create a backup such as `sample.xml-Agent-20040325`.

Remove these backup files from their respective directories and copy them to another location outside of the server directory before the SAP Enterprise Portal or Web Application Server is started for the first time after the uninstallation of the agent.

This removal process is necessary to ensure that no backup files are mistaken for configuration files by the SAP runtime. Moreover, even though the files left behind are harmless, they can cause clutter in the long run and can be difficult to remove if the SAP runtime has marked them for being stored in its database.

These backup files are located in the following directories. Where *base_instance_dir* refers to the directory in which the server instance is located:

- *base-instance-dir*
- *base-instance-dir*/ume (only if Enterprise Portal is installed)
- *base-instance-dir*/cluster/server
- *base-instance-dir*/cluster/server/managers
- *base-instance-dir*/cluster/server/services/servlet_jsp/work/jspTemp/irj/root/WEBINF (only if Enterprise Portal is installed)
- *base-instance-dir*/cluster/server/services/security
- *base-instance-dir*/cluster/server/services/security/work

NOTE Do not delete these backup files. If the agent uninstallation gets corrupted, these files are necessary for a manual restoration.

Securing Identity Manager Software With a Policy Agent

This chapter applies to deployment scenarios where a J2EE policy agent protects Sun Java™ System Identity Manager software 5.0 SP2 and establishes single sign-on (SSO) with Sun™ ONE Identity Server 6.0 and higher software.

Topics in this chapter include:

[“Deployment Facts and Considerations” on page 327](#)

[“The Policy Agent Installation” on page 328](#)

[“Configuring the Agent” on page 328](#)

[“Configuring the Identity Manager Software” on page 329](#)

[“Provisioning Users Using Identity Manager Software” on page 334](#)

[“Protecting Identity Manager Using URL Policies” on page 337](#)

[“How to Troubleshoot the Agent Configuration” on page 338](#)

Deployment Facts and Considerations

A deployment that uses a Sun ONE Identity Server policy agent to secure Sun Java System Identity Manager software provides various benefits. Foremost, this type of deployment allows single sign-on (SSO) integration between Sun Java System Identity Manager software and Sun ONE Identity Server software. Furthermore, you can use Identity Server to define URL policies that govern access to Identity Manager software web resources.

To understand this chapter, you are expected to have familiarity with Identity Manager software and the application server or J2EE container on which the Identity Manager is installed. While this chapter provides guidance for deploying Sun Java System Identity Manager software with Sun ONE Identity Server software using a policy agent, you need to consult other sources of documentation. For example, refer to other sections of this guide, especially [Chapter 2, “Installing the Agent.”](#) Moreover, refer to the Sun Java System Identity Manager software documentation that comes with that product.

The Policy Agent Installation

This section presents areas to consider about policy agent installation. Even if the agent has already been installed, this section provides important information about specific settings.

Ensure that application server you are using is supported by both Identity Manager software and by Identity Server software. For example, Sun ONE Application Server 7.0 is supported by both. A list of supported application servers for Identity Manager software, can be found in the Identity Manager documentation. For Identity server, see [“Supported Servers” on page 24.](#)

To install the agent, refer to the proper section of [Chapter 2, “Installing the Agent,”](#) for your specific agent. Settings of importance for the installation are as follows:

- For the agent filter mode, choose `SSO_ONLY`. If you later want to create URL policies, at that time, update the `AMAgent.properties` file by changing the agent filter mode to `URL_POLICY` mode.
- For the primary application context path, specify the context root for Identity Manager software.
- Skip the creation of Role-to-Principal mapping section in the install.

Configuring the Agent

After you have completed the post-installation tasks, configure the policy agent to set up an HTTP header named `sois_user`. Identity Manager uses this HTTP header to identify the authenticated user and then maps that authenticated user to a user in its repository.

To enable the policy agent to set up HTTP headers

1. Update the `AMAgent.properties` file by editing two properties as follows:
 - o LDAP Attribute Map property


```
com.sun.am.policy.amFilter.ldapattribute.map[entrydn] = sois_user
```
 - o LDAP Attribute Fetch Mode property


```
com.sun.am.policy.amFilter.ldapattribute.mode = HEADER
```
2. (Conditional) Only perform this step if the hotswap mechanism of the policy agent is disabled. If the hotswap mechanism is disabled, the Configuration Reload Interval property is set as follows:

```
com.sun.am.policy.config.load.interval = 0
```

If the hotswap mechanism is disabled, restart the application server.

Configuring the Identity Manager Software

After you have completed the installation of the policy agent, configure the Identity Manager software to allow SSO with the Sun ONE Identity Server software by performing the following general tasks:

- Adding the Resource Adapter
- Configuring the Resource Instance
- Creating a Login Module Group
- Assigning the Login Module Group to Login Applications

Adding the Resource Adapter

The `SunONEIdentityServer` resource adapter provides a login module that enables SSO between the policy agent and Identity Manager. The first task is to add the adapter, using the Identity Manager administrative console.

To add the resource adapter

1. Using an Internet browser, go to:

`http://hostname:port/idm-context-root/login.jsp`

For example:

`http://example.com:8080/idm/login.jsp`

Since the agent is already installed and configured, the Identity Server Login Page displays in the browser.

2. Enter a valid user ID and password. For example, the credentials for amAdmin or any Identity Server software user work for this step.

The Identity Manager administrative login page appears.

3. Log in as Identity Manager Configurator user.

The Identity Manager administrative console appears.

4. Select the Configure tab.

5. In the left-hand menu, select Managed Resources.

The Configure Managed Resources page appears.

6. Under the Custom Resources heading and the Resource Class Path subheading, add the following string:

```
com.waveset.adapter.SunISResourceAdapter
```

7. Click Add Custom Resource.

8. Click Save.

Configuring the Resource Instance

At this point, the resource adapter has been added.

To configure a resource instance

1. Using the Identity Manager administrative console, select the Resources tab.

2. In the left-hand menu, select List Resources.

A box appears that contains a directory tree of resources.

3. Under the box, in the drop-down list, select Sun ONE Identity Server if it is not already selected.
4. If necessary, click Edit.
The SunONEIdentityServer resource wizard appears.
5. Click Next.
The Resource Parameters page appears.
6. Enter the details for the directory server. Here, *directory server* refers to the directory server used by the Identity Server software to which the policy agent will communicate. Enter the following information.

Resource	Parameters
Host	<i>DirectoryServer-hostname</i>
TCP Port	<i>DirectoryServer-port</i>
User	amAdmin
Password	<i>amAdmin-password</i>
Base Context	<i>IdentityServer-root-suffix</i>

7. Click Test Configuration to test for a successful connection.
 - o If the test was successful, continue.
 - o If the test was unsuccessful, return to [Step 6](#), and enter the correct information.
8. Click Next.
The Account Attributes page of the wizard appears.
9. Select the defaults.
10. Click Next.
The Identity Template page of the wizard appears.
11. Update the template by entering information in the Identity Template field that reflects a user DN of Identity Server.

For example:

```
uid=$uid$,ou=People,dc=example,dc=com
```

12. Click Next.

The Identity Manager Parameters page appears.

13. In the Display Name Attribute drop-down list, select a display name. For example, firstname.
14. Locate the item named Top inside the Organizations box and move it to the Available To box.
15. For the other options on this page, keep the defaults or update the information according to your requirements.
16. Click Next

Creating a Login Module Group

To create a new Identity Server software login module group

1. Using the Identity Manager administrative console, select the Configure tab.

2. In the left-hand menu, select Login.

The Login page appears.

3. Click Manage Login Module Groups.

The Login Module Groups page appears.

4. Click New.

The Create Login Module Group page appears.

5. In the Login Manager Group Name field, enter a name for the new login module group.

For example:

Access Manager Login Module Group

6. Locate the item Top inside the Organizations box and move it to the Available To box.
7. From the Assign Login Module drop down list, select Sun ONE Identity Server Login Module.

The Create Login Module Group page reappears, but with a new drop-down list available.

8. In the newly available drop-down list, select SunONEIdentityServer.
The Modify Login Group page appears.
9. From the Login success requirement drop-down list, select sufficient.
10. Click Save.
The Create Login Module Group page reappears.
11. In the Assign Login Module Groups drop-down list, select the Identity Manager User Id/Password Login Module.
The Modify Login Module page appears.
12. In the Login success requirement drop-down list, select sufficient.
13. Click Save.
The Create Login Module page reappears with a list of two login modules:
 - o The Sun ONE Identity Server Login module
 - o The Identity Manager User Id/Password Login module
14. Click Save to save the configuration.
15. Verify the creation of the Identity Server Login Module group in the Manage Login Module Group page, which you can access by repeating steps 1—3.

NOTE Assigning both modules, Identity Server Login module and Identity Manager User Id/Password Login module increases the availability of protected resources. Thereafter, even if the Sun ONE Identity Server Login module fails, users can log in to the Identity Manager software using the Identity Manager User Id/Password Login module.

Assigning the Login Module Group to Login Applications

You need to configure two interface login applications, the User Interface Login Application and the Administrative Interface Login Application.

Configuring the User Interface Login Application

1. Using the Identity Manager administrative console, click the Configure tab.

2. In the left-hand menu, click Login.
The Login page appears.
3. Click the User Interface login application.
The Modify Login Application page appears.
4. Select the newly created Identity Server Login Module Group from the Assign Login Module Groups drop-down list.
The Modify Login Application page reappears with the newly created group added to the list of login module groups.
5. Remove the old Login Module group from the list as follows:
 - a. Select that module
 - b. Click delete.
 - c. Click Save.

The new login module group for the user interface login application has now been configured.

Configuring the Administrative Interface Login Application

The steps to assign the Identity Server Login Module Group to Administrative Interface are the same as those described in [“Configuring the User Interface Login Application” on page 333](#), except instead of choosing User Interface Login Application, choose the Administrator Interface login application.

Provisioning Users Using Identity Manager Software

To provision users using Identity Manager software

1. Using the Identity Manager administrative console, click the Accounts tab.
2. In the left-hand menu, select List Accounts if it is not already selected.
A box appears that contains user names.
3. Select a user name:
Select an existing Identity Manager software user or create a new user:

- For an existing Identity Manager software user, perform the following:
 - a. Select the user.
 - b. Select Edit.
- For a new Identity Manager software user, perform the following:

Select User from the New drop down list.

After you perform one of the preceding tasks, the Create User or Edit User page appears. These pages have four tabs: Identity, Assignments, Security, and Attributes. You can navigate between these tabs without losing newly added data. However, select Save after all information is added.

4. Select the Assignments tab.
5. Locate the item SunONEIdentityServer inside the Available Resources list box and move it to the Current Resources list box.
6. Select the Attributes tab.
7. Enter the value for the uid (The uid is called user ID in Identity Server software).
8. (Conditional) This step only applies if you are creating a new user. Select the Identity tab.
9. (Conditional) This step only applies if you are creating a new user. Specify the user Account ID and password.
10. Click Save.
11. (Conditional) This step only applies if you are editing an existing user. In which case, the Update *user_name* Resource Accounts page is currently available in your browser. Review the information about the accounts that will be updated and click Save.
12. Now that you are in the Create User Results page or the Update Resource Account Results page, click OK.

You are returned to the List Accounts page.

After successfully creating or updating a user, perform a simple test as explained in [“Testing SSO Integration With Identity Server” on page 336](#) to verify that SSO functions properly.

NOTE When managing users who were created using Identity Server software, ensure that passwords are synchronized through Identity Manager software.

Testing SSO Integration With Identity Server

Testing SSO for users with administrative capabilities differs slightly from testing SSO for users without administrative capabilities as explained in the following subsections.

Testing SSO for Users With Administrative Capabilities

The best method for testing SSO is to keep your original browser instance open. This method maintains user sessions. Therefore, keep your browser connected to the Identity Manager administrative console. Then open a new browser window using a different browser (for example: open a new Mozilla/Firefox instance if you are currently using the IE instance).

1. Using an Internet browser, go to:

`http://hostname:port/application-context-root/login.jsp`

For example:

`http://example.com:8080/idm/login.jsp`

The browser presents the Identity Server Login Page.

2. Enter the user ID and password for the newly created user.

You should be presented with the Identity Manager administrative console.

If you are able to directly access the Identity Manager software Administrator page, then SSO is working and the test is successful.

Testing SSO for Users Without Administrative Capabilities

The best method for testing the SSO connection is to keep your original browser instance open. This method maintains user sessions. Therefore, keep your browser connected to the Identity Manager administrative console. Then open a new browser window using a different browser (for example: open a new Mozilla/Firefox instance if you are currently using the IE instance).

1. Using an Internet browser, go to:

`http://hostname:port/application-context-root/user/login.jsp`

For example:

`http://example.com:8080/idm/user/login.jsp`

The browser presents the Identity Server Login Page.

2. Enter the user ID and password for the newly created user.

The Identity Manager user profile page appears.

If you are able to directly access the user profile page, which displays links to Change Password and Change Other Account Attributes, then SSO is working and the test is successful.

Protecting Identity Manager Using URL Policies

Depending upon your environment needs, you might want to use Identity Server software to define URL policies that protect Identity Manager software. For example, you could assign authorization to specific users to access Identity Manager software.

The following procedure is an overview of the steps for creating a URL Policy. However, consult the administration guide for Sun ONE Identity Server software or the administration guide for Sun Java™ System Access Manager for detailed information about creating policies.

To create a simple URL policy based enforcement

1. Log in to the Identity Server software as Administrator.
2. Create a new policy.

The following are guidelines for that policy

- Give the policy a relevant name, such as IDMGR.
- Apply the following rules:
 - `http://hostname:port/application-context-root`

where *hostname* and *port* refer to the Identity Manager software

where actions are set to allow GET and POST commands.

- `http://hostname:port/application-context-root/*`
where *hostname* and *port* refer to the Identity Manager software
where actions are set to allow GET and POST commands.
3. Assign one or more subjects to the IDMGR policy.
 4. Set the Agent Filter Mode property in the `AMAgent.properties` file to the following:

```
com.sun.am.policy.amFilter.mode = URL_POLICY
```
 5. (Conditional) If the hotswap mechanism is disabled, restart the application server.

How to Troubleshoot the Agent Configuration

If the agent does not operate as expected, you can follow the troubleshooting steps described in this section. You can set trace options to help you troubleshoot problems in Identity Manager software. However, refer to the *Identity Manager Technical Reference* for detailed information.

1. Use the Identity Manager software debug pages to set trace options on the following class:

```
com.waveset.adapter.SunISResourceAdapter
```
2. Also, set the debug level to “message” in the `amagent.properties`
3. Restart the application Server
4. Inspect the debug logs of the Policy Agent for troubleshooting information.

Silent Installation

In addition to the GUI installation and the Command Line installation, you can do a silent installation of J2EE Policy Agents. This appendix provides steps for doing that.

About Silent Installation/Uninstallation

Silent installation/uninstallation provides a means for scripting the installation/uninstallation. When you perform a silent installation/uninstallation, you use a state file, to provide all the answers that you would normally supply to the installation/uninstallation program interactively. This saves time and is useful when you want to install/uninstall multiple instances of Sun ONE Identity Server Policy Agent using the same parameters in each instance.

Silent installation is a simple two-step process. First, you generate a state file that records the installation/uninstallation process and all the inputs you provide. Then you run the installation/uninstallation program with the state file as the input source.

Generating a State File for Installation

To generate a state file, do the following:

1. Change to the directory where the installation program is located.
2. Enter one of the following commands if you are running the GUI installation program.

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
# ./setup.sh -saveState StateFile
```

If you are on Windows, enter:

```
C:\> setup.bat -saveState StateFile
```

You can give a name of your choice to the state file. Follow the instructions as specified in the section [“Using the GUI Installation Program.”](#)

3. Alternatively, you can enter one of the following commands if you are using the command line installation program.

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
# ./setup.sh -nodisplay -saveState stateFile
```

If you are on Windows, enter:

```
C:\> setup.bat -nodisplay -saveState stateFile
```

You can give a name of your choice to the StateFile. Follow the instructions as specified in the section [“Using the Command-Line Installation Program.”](#)

4. Proceed through the installation. 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, it is recommended that you change the file permissions to restrict read/write access to the user root.

Using the State File for Silent Installation

To perform silent installation using the state file, do the following.

1. Change to the directory where the installation program and the state file are located.
2. Enter the following command:

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
# ./setup.sh -nodisplay -noconsole -state StateFile
```

If you are on Windows, enter:

```
C:\> setup.bat -nodisplay -noconsole -state StateFile
```

The installation takes place hidden from your view. On completion, the program exits automatically and displays the prompt.

NOTE The installation program doesn't validate inputs or the state in the silent installation. You must ensure that the proper environment exists before performing the silent installation.

Generating a State File for Uninstallation

To generate a state file for uninstallation, do the following.

1. Change to the agent installation directory, where the uninstallation program is located.
2. Enter one of the following commands if you are using the GUI installation program:

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
# ./uninstall.sh -saveState StateFile
```

If you are on Windows, enter:

```
C:\> uninstall.bat -saveState StateFile
```

You can give a name of your choice to the state file. Follow the instructions as specified in the section [“Using the GUI Uninstallation Program” on page 312](#) to proceed with the uninstallation.

3. Alternatively, you can enter one the following commands if you are using the command line installation program.

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
#./uninstall.sh -nodisplay -saveState stateFile
```

If you are on Windows, enter:

```
C:\> uninstall.bat -nodisplay -saveState stateFile
```

You can give a name of your choice to the state file.

4. Proceed through the uninstallation program as explained in [“Using the GUI Uninstallation Program” on page 312](#). Your answers to the prompts are recorded in the state file. When uninstallation is complete, the state file is created in the same directory where the uninstallation 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, it is recommended that you change the file permissions to restrict read/write access to the user `root`.

Using the State File for Silent Uninstallation

To perform silent uninstallation using the state file, do the following:

1. Change to the agent installation directory where uninstallation program and the state file are located.
2. Update the state file with the ID of the uninstallation program class as follows:

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
#./uninstall.sh -id
```

If you are on Windows, enter:

```
C:\> uninstall.bat -id
```

This command will print the id of the uninstallation program class file.

3. Replace the ID in the state file with this ID.

NOTE The state file id for the uninstallation program will change every time because the installation program creates a new class file for uninstallation. While running uninstallation program in the silent mode, you must ensure that the id in the class file matches the id of the uninstallation class.

4. Begin the uninstallation process by using one of the following commands:

If you are on Solaris, HP-UX, AIX, or Linux, enter:

```
# ./uninstall.sh -nodisplay -noconsole -state StateFile
```

If you are on Windows, enter:

```
C:\> uninstall.bat -nodisplay -noconsole -state StateFile
```

The uninstallation takes place hidden from your view. On completion, the program exits automatically and displays the prompt.

NOTE The uninstallation program doesn't validate inputs or the state in the silent uninstallation. The user must take care to ensure that the proper environment exists before performing the silent mode uninstallation

Sample Application Scenario

This appendix is divided into two broad sections:

- [Standard Deployment Descriptors](#)
- [Assembly Descriptors](#)

The sample application presented in this section consists of a web module and an EJB Module.

The web module consists of the following two servlets:

- the `SummaryReportServlet`, which can be accessed by both employees and managers
- the `DetailedReportServlet`, which can be accessed by only managers.

The EJB Module consists of a single EJB with two methods:

- `doSummaryReport()` that can be accessed by both employees and managers
- `doDetailedReport()` that can be accessed by only managers.

Standard Deployment Descriptors

The following deployment descriptors are common to all J2EE servers:

- `web.xml`
- `ejb-jar.xml`
- `application.xml`

web.xml

Code Example B-1 web.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app id="WEBAPP_REPORT">
  <display-name>Sample Report Application</display-name>
  <description>Sample Report Application to illustrate J2EE Security</description>
  <filter id="filter_agent">
    <filter-name>AgentFilter</filter-name>
    <filter-class>Filter Class</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>AgentFilter</filter-name>
    <url-pattern>*/</url-pattern>
  </filter-mapping>
  <servlet id="Servlet_SummaryReport">
    <servlet-name>SummaryReportServlet</servlet-name>
    <servlet-class>com.example.reportapp.SummaryReportServlet</servlet-class>
  </servlet>
  <servlet id="Servlet_DetailedReport">
    <servlet-name>DetailedReportServlet</servlet-name>
    <servlet-class>com.example.reportapp.DetailedReportServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>SummaryReportServlet</servlet-name>
    <url-pattern>/SummaryReportServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>DetailedReportServlet</servlet-name>
    <url-pattern>/DetailedReportServlet</url-pattern>
  </servlet-mapping>

  <security-constraint>
    <web-resource-collection>
      <url-pattern>/DetailedReportServlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>MANAGER-ROLE</role-name>
    </auth-constraint>
  </security-constraint>
  <security-constraint>
    <web-resource-collection>
      <url-pattern>/SummaryReportServlet</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>MANAGER-ROLE</role-name>
      <role-name>EMPLOYEE-ROLE</role-name>
    </auth-constraint>
  </security-constraint>
  <login-config>
    <auth-method>FORM</auth-method>
  </login-config>

```

Code Example B-1 web.xml

```
<form-login-config>
  <form-login-page>/login.html</form-login-page>
  <form-error-page>/deny.html</form-error-page>
</form-login-config>
</login-config>

<security-role id="ID_MANAGER_ROLE">
  <role-name>MANAGER-ROLE</role-name>
</security-role>
<security-role id="ID_EMPLOYEE_ROLE">
  <role-name>EMPLOYEE-ROLE</role-name>
</security-role>
</web-app>
```

NOTE**Concerning the Filter Class in the web.xml File**

- The filter class for Sun ONE Application Server 7.0, BEA WebLogic Server 6.1 SP2, BEA WebLogic Server 7.0 SP2, BEA WebLogic Server 8.1 and Macromedia JRun 4 is as follows:
com.sun.identity.agents.filter.AmAgentFilter.
- The following filter class applies to both IBM WebSphere Application Server 5.0 and 5.1:
com.sun.identity.agents.websphere.AmWAS50AgentFilter.

NOTE**Concerning the Dispatcher Element in the web.xml File**

If you are using Sun Java System Application Server 8.1 as your application server, be aware that this application server supports the Java Servlet Specification version 2.4, which requires you to update the standard deployment descriptors for the Agent for Sun Java System Application Server 8.1.

A new dispatcher element has been added to the `filter-mapping` element in servlet 2.4. The agent filter requires the dispatcher element with values `REQUEST`, `FORWARD`, `INCLUDE`, and `ERROR` in the Sun Java System Application Server 8.1 Agent. See the Post Installation Tasks section, specifically [“Agent for Sun Java System Application Server 8.1” on page 149](#) for more information.

The following is a sample Servlet 2.4 compliant web.xml descriptor with the `<filter>`, `<filter-mapping>`, and `<dispatcher>` elements added:

```
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>...</display-name>
  <description>...</description>
  <filter>
    <filter-name>Agent</filter-name>
    <display-name>Agent</display-name>
    <description>Sun Java System Access Manager Policy Agent Filter
</description>
    <filter-class>com.sun.identity.agents.filter.AmAgentFilter</filter-class>
  </filter>
  <filter-mapping id="FilterMapping_PolicyAgent">
    <filter-name>Agent</filter-name>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>INCLUDE</dispatcher>
    <dispatcher>FORWARD</dispatcher>
    <dispatcher>ERROR</dispatcher>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  ...
  ...
</web-app>
```

ejb-jar.xml

Code Example B-2 ejb-jar.xml

```

<?xml version="1.0"?>
<!DOCTYPE ejb-jar PUBLIC
  "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN"
  "http://java.sun.com/dtd/ejb-jar_2_0.dtd">
<ejb-jar id="ID_ejb_jar">
  <enterprise-beans>
    <session id="Session_ReportEJB">
      <ejb-name>ReportEJB</ejb-name>
      <home>com.example.reportapp.ReportEJBHome</home>
      <remote>com.example.reportapp.ReportEJB</remote>
      <ejb-class>com.example.reportapp.ReportEJBImpl</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
  <assembly-descriptor>
    <security-role id="ID_MANAGER_ROLE">
      <role-name>MANAGER-ROLE</role-name>
    </security-role>
    <security-role id="ID_EMPLOYEE_ROLE">
      <role-name>EMPLOYEE-ROLE</role-name>
    </security-role>
    <method-permission>
      <role-name>MANAGER-ROLE</role-name>
      <method>
        <ejb-name>ReportEJB</ejb-name>
        <method-intf>Remote</method-intf>
        <method-name>doDetailedReport</method-name>
        <method-params/>
      </method>
    </method-permission>
    <method-permission>
      <role-name>MANAGER-ROLE</role-name>
      <role-name>EMPLOYEE-ROLE</role-name>
      <method>
        <ejb-name>ReportEJB</ejb-name>
        <method-intf>Remote</method-intf>
        <method-name>doSummaryReport</method-name>
        <method-params/>
      </method>
    </method-permission>
  </assembly-descriptor>
</ejb-jar>

```

application.xml

Code Example B-3 application.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE application PUBLIC
    "-//Sun Microsystems, Inc.//DTD J2EE Application 1.3//EN"
    "http://java.sun.com/dtd/application_1_3.dtd" >
<application id="APP_REPORTAPP">
    <display-name>Sample Report Application</display-name>
    <description>Sample Report Application to illustrate J2EE
    Security</description>
    <module id="em_reportapp">
        <ejb>reportapp.jar</ejb>
    </module>
    <module id="wm_reportapp">
        <web>
            <web-uri>reportapp.war</web-uri>
            <context-root>ReportApp</context-root>
        </web>
    </module>
    <security-role id="ID_MANAGER_ROLE">
        <role-name>MANAGER-ROLE</role-name>
    </security-role>

    <security-role id="ID_EMPLOYEE_ROLE">
        <role-name>EMPLOYEE-ROLE</role-name>
    </security-role>
</application>

```

Assembly Descriptors

This section provides samples of descriptors for:

- [Sun ONE Application Server 7.0](#)
- [BEA WebLogic 6.1 SP2](#)
- [IBM WebSphere Application Server 5.0/5.1](#)
- [BEA WebLogic Server 7.0 SP2 and BEA WebLogic Server 8.1](#)
- [Apache Tomcat Server 4.1.27](#)
- [Macromedia JRun 4](#)
- [Oracle 9iAS R2 and Oracle 10g](#)

- [Sun Java System Application Server 8.1](#)

Sun ONE Application Server 7.0

The following are samples of deployment descriptors to support Declarative (at Web and EJB tier) and Programmatic security for Sun One Application Server 7.0:

- [sun-web.xml](#)
- [sun-ejb-jar.xml](#)
- [sun-application.xml](#)

These samples are not exhaustive and the user should refer to Sun ONE Application Server reference documentation to learn the details on how these descriptors may be edited to create the role-to-principal mappings:

<http://docs.sun.com/source/816-7149-10/index.html>

sun-web.xml

Code Example B-4 sun-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Sun ONE
Application Server 7.0 Servlet 2.3//EN"
'http://www.sun.com/software/sunone/appserver/dtds/sun-web-app_2_3-0.dtd'>
<!-- Copyright 2002 Sun Microsystems, Inc. All rights reserved. -->
<sun-web-app>
  <security-role-mapping>
    <role-name>MANAGER-ROLE</role-name>
    <group-name>manager</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>EMPLOYEE-ROLE</role-name>
    <group-name>employee</group-name>
  </security-role-mapping>
</sun-web-app>
```

sun-ejb-jar.xml

Code Example B-5 sun-ejb-jar.xml

```
<?xml version="1.0"?>
<!DOCTYPE sun-ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Sun ONE
Application Server 7.0 EJB 2.0//EN"
'http://www.sun.com/software/sunone/appserver/dtds/sun-ejb-jar_2_0-0.dtd'>

<sun-ejb-jar>
  <enterprise-beans>
    <ejb>
      <ejb-name>ReportEJB </ejb-name>
      <jndi-name>ReportEJB </jndi-name>
    </ejb>
  </enterprise-beans>
</sun-ejb-jar>
```

sun-application.xml

Code Example B-6 sun-application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-application PUBLIC "-//Sun Microsystems, Inc.//DTD Sun ONE
Application Server 7.0 J2EE Application 1.3//EN"
'http://www.sun.com/software/sunone/appserver/dtds/sun-application_1_3-0.dtd'>
<sun-application>
  <security-role-mapping>
    <role-name>MANAGER-ROLE</role-name>
    <group-name>manager</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>EMPLOYEE-ROLE</role-name>
    <group-name>employee</group-name>
  </security-role-mapping>
</sun-application>
```

BEA WebLogic 6.1 SP2

The following are samples of deployment descriptors to support declarative (at Web and EJB tier) and programmatic security for BEA WebLogic 6.1SP2.

- [weblogic.xml](#)
- [weblogic-ejb-jar.xml](#)

These samples are not exhaustive and the user should refer to the BEA WebLogic 6.1SP2 reference documentation to learn the details on how these descriptors may be edited to create the role-to-principal mappings:

<http://e-docs.bea.com/wls/docs61/programming.html>

weblogic.xml

Code Example B-7 weblogic.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application
6.0//EN" "http://www.bea.com/servers/wls600/dtd/weblogic-web-jar.dtd">
<weblogic-web-app>
  <description>WebLogic Descriptor</description>
  <security-role-assignment>
    <role-name>MANAGER-ROLE</role-name>
    <principal-name>manager</principal-name>
  </security-role-assignment>
  <security-role-assignment>
    <role-name>EMPLOYEE-ROLE</role-name>
    <principal-name>employee</principal-name>
  </security-role-assignment>
</weblogic-web-app>
```

weblogic-ejb-jar.xml

Code Example B-8 weblogic-ejb-jar.xml

```
<?xml version="1.0"?>

<!DOCTYPE weblogic-ejb-jar PUBLIC "-//BEA Systems, Inc.//DTD WebLogic 6.0.0
EJB//EN" "http://www.bea.com/servers/wls600/dtd/weblogic-ejb-jar.dtd">

<weblogic-ejb-jar>
  <weblogic-enterprise-bean>
    <ejb-name>ReportEJB </ejb-name>
    <jndi-name>ReportEJB </jndi-name>
  </weblogic-enterprise-bean>
  <security-role-assignment>
    <role-name>MANAGER-ROLE</role-name>
    <principal-name>manager</principal-name>
  </security-role-assignment>
  <security-role-assignment>
    <role-name>EMPLOYEE-ROLE</role-name>
    <principal-name>employee</principal-name>
  </security-role-assignment>
</weblogic-ejb-jar>
```

IBM WebSphere Application Server 5.0/5.1

The following are samples of deployment descriptors to support declarative (at Web and EJB tier) and programmatic security for IBM WebSphere Application Server 5.0/5.1:

- [ibm-web-bnd.xmi](#)
- [ibm-web-ext.xmi](#)
- [ibm-ejb-jar-bnd.xmi](#)
- [ibm-ejb-jar-ext.xmi](#)
- [ibm-application-bnd.xmi](#)
- [ibm-application-ext.xmi](#)

These samples are not exhaustive and the user should refer to the IBM WebSphere Application Server 5.0/5.1 reference documentation to learn how these descriptors may be edited to create the role-to-principal mappings:

<http://publib7b.boulder.ibm.com/webapp/wasinf01/index.jsp?deployment=ApplicationServer&lang=en>

ibm-web-bnd.xmi

Code Example B-9 bm-web-bnd.xmi

```
<?xml version="1.0" encoding="UTF-8"?>

<webappbnd:WebAppBinding xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:webappbnd="webappbnd.xmi"
  xmlns:webapplication="webapplication.xmi"
  xmi:id="WebAppBinding_1"
  virtualHostName="default_host">
  <webapp href="WEB-INF/web.xml#WEBAPP_REPORT" />
</webappbnd:WebAppBinding>
```

ibm-web-ext.xmi

Code Example B-10 ibm-web-ext.xmi

```

<?xml version="1.0" encoding="UTF-8"?>

<webappext:WebAppExtension xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:webappext="webappext.xmi"
  xmlns:webapplication="webapplication.xmi"
  xmi:id="WebAppExtension_1">

  <webApp href="WEB-INF/web.xml#WEBAPP_REPORT"/>

  <extendedServlets xmi:id="Servlet_SummaryReport_Ext">
    <extendedServlet href="WEB-INF/web.xml#Servlet_SummaryReport"/>
  </extendedServlets>

  <extendedServlets xmi:id="Servlet_DetailedReport_Ext">
    <extendedServlet href="WEB-INF/web.xml#Servlet_DetailedReport"/>
  </extendedServlets>

</webappext:WebAppExtension>

```

ibm-ejb-jar-bnd.xmi

Code Example B-11 ibm-ejb-jar-bnd.xmi

```

<?xml version="1.0" encoding="UTF-8"?>

<ejbbnd:EJBJarBinding xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:ejbbnd="ejbbnd.xmi"
  xmlns:ejb="ejb.xmi"
  xmi:id="EJBJarBinding_1">

  <ejbJar href="META-INF/ejb-jar.xml#ID_ejb_jar"/>

  <ejbBindings xmi:id="Session_ReportEJB_Bnd" jndiName="ReportEJB">
    <enterpriseBean xmi:type="ejb:Session"
      href="META-INF/ejb-jar.xml#Session_ReportEJB"/>
  </ejbBindings>

</ejbbnd:EJBJarBinding>

```

ibm-ejb-jar-ext.xmi

Code Example B-12 ibm-ejb-jar-ext.xmi

```

<?xml version="1.0" encoding="UTF-8"?>

<ejbext:EJBJarExtension xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:ejbext="ejbext.xmi"
  xmlns:ejb="ejb.xmi"
  xmi:id="EJBJarExtension_1">

  <ejbExtensions xmi:type="ejbext:SessionExtension"
    xmi:id="SessionExtension_1"
    name="SessionExtension_1">
    <enterpriseBean xmi:type="ejb:Session"
      href="META-INF/ejb-jar.xml#Session_ReportEJB"/>
  </ejbExtensions>

  <ejbJar href="META-INF/ejb-jar.xml#ID_ejb_jar"/>
</ejbext:EJBJarExtension>

```

ibm-application-bnd.xmi

Code Example B-13 ibm-application-bnd.xmi

```

<?xml version="1.0" encoding="UTF-8"?>

<applicationbnd:ApplicationBinding xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:applicationbnd="applicationbnd.xmi"
  xmlns:common="common.xmi"
  xmlns:application="application.xmi"
  xmi:id="ApplicationBinding_1">

  <authorizationTable xmi:id="AuthorizationTable_1">
    <authorizations xmi:id="RoleAssignment_1">
      <role href="META-INF/application.xml#ID_EMPLOYEE_ROLE"/>
      <groups xmi:id="Group_1" name="employee"/>
    </authorizations>
    <authorizations xmi:id="RoleAssignment_2">
      <role href="META-INF/application.xml#ID_MANAGER_ROLE"/>
      <groups xmi:id="Group_2" name="manager"/>
    </authorizations>
  </authorizationTable>

  <application href="META-INF/application.xml#APP_REPORTAPP"/>
</applicationbnd:ApplicationBinding>

```

ibm-application-ext.xmi

Code Example B-14 ibm-application-ext.xmi

```
<?xml version="1.0" encoding="UTF-8"?>

<applicationext:ApplicationExtension xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI"
  xmlns:applicationext="applicationext.xmi"
  xmlns:application="application.xmi"
  xmi:id="ApplicationExtension_1">

  <moduleExtensions xmi:type="applicationext:EjbModuleExtension"
    xmi:id="EjbModuleExtension_1"
    altRoot="ALT-INF/reportapp.jar">

    <module xmi:type="application:EjbModule"
      href="META-INF/application.xml#em_reportapp"/>

  </moduleExtensions>

  <moduleExtensions xmi:type="applicationext:WebModuleExtension"
    xmi:id="WebModuleExtension_1"
    altRoot="ALT-INF/reportapp.war">
    <module xmi:type="application:WebModule"
      href="META-INF/application.xml#wm_reportapp"/>
  </moduleExtensions>
  <application href="META-INF/application.xml#APP_REPORTAPP"/>
</applicationext:ApplicationExtension>
```

BEA WebLogic Server 7.0 SP2 and BEA WebLogic Server 8.1

The following are samples of deployment descriptors to support declarative (at Web and EJB tier) and programmatic security for BEA WebLogic Server 7.0 SP2 or BEA WebLogic Server 8.1:

- [weblogic.xml](#)
- [weblogic-ejb-jar.xml](#)

These samples are not exhaustive and the user should refer to the BEA WebLogic Server reference documentation to learn how these descriptors may be edited to create the role-to-principal mappings.

BEA WebLogic Server 7.0 related documentation:

<http://e-docs.bea.com/wls/docs70/programming.html>

BEA WebLogic Server 8.1 related documentation

<http://e-docs.bea.com/wls/docs81/index.html>

weblogic.xml

Code Example B-15 weblogic.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application
7.0//EN" "http://www.bea.com/servers/wls700/dtd/weblogic700-web-jar.dtd">

<weblogic-web-app>
  <description>WebLogic Descriptor</description>
  <security-role-assignment>
    <role-name>MANAGER-ROLE</role-name>
    <principal-name>manager</principal-name>
  </security-role-assignment>
  <security-role-assignment>
    <role-name>EMPLOYEE-ROLE</role-name>
    <principal-name>employee</principal-name>
  </security-role-assignment>
</weblogic-web-app>
```

weblogic-ejb-jar.xml

Code Example B-16 weblogic-ejb-jar.xml

```
<?xml version="1.0"?>

<!DOCTYPE weblogic-ejb-jar PUBLIC "-//BEA Systems, Inc.//DTD WebLogic 7.0.0
EJB//EN" 'http://www.bea.com/servers/wls700/dtd/weblogic-ejb-jar.dtd'>

<weblogic-ejb-jar>
  <weblogic-enterprise-bean>
    <ejb-name>ReportEJB </ejb-name>
    <jndi-name>ReportEJB </jndi-name>
  </weblogic-enterprise-bean>

  <security-role-assignment>
    <role-name>MANAGER-ROLE</role-name>
    <principal-name>manager</principal-name>
  </security-role-assignment>
  <security-role-assignment>
    <role-name>EMPLOYEE-ROLE</role-name>
    <principal-name>employee</principal-name>
```


Code Example B-16 weblogic-ejb-jar.xml

```

    </security-role-assignment>
</weblogic-ejb-jar>

```

Apache Tomcat Server 4.1.27

The following is a sample of deployment descriptors to support Declarative and Programmatic security for Apache Tomcat Server 4.1.27.

- [web.xml](#)

Tomcat Server 4.1.27 does not include an EJB Container hence there is no support for EJBs. All security constraints and role-to-principal mappings are contained within the web application descriptor file. A separate role-to-principal mapping file is not needed for Tomcat Server. These samples are not exhaustive and the user should refer to the standard J2EE security guide at the following location for additional details.

http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Security.html

web.xml

Code Example B-17 web.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    <servlet id="SampleServlet">
        <servlet-name>SampleServlet</servlet-name>
        <display-name>SampleServlet</display-name>
        <description>A Sample Servlet</description>
        <servlet-class>test.SampleServlet</servlet-class>
        <security-role-ref>
            <role-name>MANAGER-ROLE</role-name>
            <role-link>manager</role-link>
        </security-role-ref>
    </servlet>
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>Protected</web-resource-name>
            <url-pattern>*.jsp</url-pattern>

```

Code Example B-17 web.xml

```

        <url-pattern>*.html</url-pattern>
    </web-resource-collection>
    <auth-constraint>
    <!-- Anyone with one of the listed roles may access this area -->
        <role-name>manager</role-name>
    </auth-constraint>
</security-constraint>

    <security-role id="MANAGER-ROLE">
        <description> The role that is required to access this sample
        application </description>
        <role-name>manager</role-name>
    </security-role>

</web-app>

```

Macromedia JRun 4

The standard deployment descriptors defined in the section [Standard Deployment Descriptors](#) to support Declarative and Programmatic security will work for Macromedia JRun 4 and a separate role-to-principal mapping file is not required.

Oracle 9iAS R2 and Oracle 10g

The following are samples of deployment descriptors to support declarative (at Web and EJB tier) and programmatic security for Oracle 9iAS R2 and Oracle 10g.

- [orion-web.xml](#)
- [orion-ejb-jar.xml](#)
- [orion-application.xml](#)

These samples are not exhaustive and the user should refer to Oracle documentation to learn the details on how these descriptors may be edited to create the role-to-principal mappings:

<http://otn.oracle.com/documentation/ias.html>

orion-web.xml

Code Example B-18 orion-web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE orion-web-app PUBLIC "-//Evermind//DTD Orion Web Application
2.3//EN" "http://xmlns.oracle.com/ias/dtds/orion-web.dtd">

<orion-web-app deployment-version="1.0.2.2"
  jsp-cache-directory="./persistence"
  temporary-directory="./temp"
  servlet-webdir="/servlet/">
</orion-web-app>
```

orion-ejb-jar.xml

Code Example B-19 orion-ejb-jar.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE orion-ejb-jar PUBLIC "-//Evermind//DTD Enterprise JavaBeans 1.1
runtime//EN" "http://www.orionserver.com/dtds/orion-ejb-jar.dtd">
<orion-ejb-jar>
  <enterprise-beans>
    <session-deployment name="ReportEJB" location="ReportEJB" />
  </enterprise-beans>
</orion-ejb-jar>
```

orion-application.xml

Code Example B-20 orion-application.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE orion-application PUBLIC "-//Evermind//DTD J2EE Application
runtime 1.2//EN" "http://xmlns.oracle.com/ias/dtds/orion-application.dtd">

<orion-application deployment-version="1.0.2.2">
  <web-module id="WebTestSuite" path="War_File_Name.war" />
  <ejb-module id="EJBTestSuite" path="EJB_File_Name.jar" />

  <security-role-mapping name="MANAGER-ROLE">
    <group name="manager" />
  </security-role-mapping>

  <security-role-mapping name="EMPLOYEE-ROLE">
    <group name="employee" />
  </security-role-mapping>
</orion-application>
```

Code Example B-20 orion-application.xml

```

</security-role-mapping>

<persistence path="persistence" />
<user-manager
class="com.sun.identity.agents.oracle.AmOracleUserManager">
</user-manager>

<log>
<file path="application.log" />
</log>

<namespace-access>
<read-access>
<namespace-resource root="">
<security-role-mapping name="&lt;jndi-user-role">">
<group name="administrators" />
</security-role-mapping>
</namespace-resource>
</read-access>
<write-access>
<namespace-resource root="">
<security-role-mapping name="&lt;jndi-user-role">">
<group name="administrators" />
</security-role-mapping>
</namespace-resource>
</write-access>
</namespace-access>
</orion-application>

```

NOTE If the user does not add the custom User Manager tag, it will be added by default when deploying the application through Enterprise Manager.

Sun Java System Application Server 8.1

The following are samples of deployment descriptors to support Declarative (at Web and EJB tier) and Programmatic security for Sun Java System Application Server 8.1:

- [sun-web.xml](#)
- [sun-ejb-jar.xml](#)
- [sun-application.xml](#)

Update the DTD files in each of these deployment descriptors for Sun Java System Application Server 8.1. Keep in mind, that these samples are not exhaustive and the user should refer to *Sun Java System Application Server Developer's Guide* to learn the details of how these descriptors can be edited to create the role-to-principal mappings.

sun-web.xml

Code Example B-21 sun-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Application
Server 8.1 Servlet 2.4//EN"
'http://www.sun.com/software/appserver/dtds/sun-web-app_2_4-1.dtd'>
<!-- Copyright 2002 Sun Microsystems, Inc. All rights reserved. -->
<sun-web-app>
  <security-role-mapping>
    <role-name>MANAGER-ROLE</role-name>
    <group-name>manager</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>EMPLOYEE-ROLE</role-name>
    <group-name>employee</group-name>
  </security-role-mapping>
</sun-web-app>
```

sun-ejb-jar.xml

Code Example B-22 sun-ejb-jar.xml

```
<?xml version="1.0"?>
<!DOCTYPE sun-ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD
Application Server 8.1 EJB 2.1//EN"
'http://www.sun.com/software/appserver/dtds/sun-ejb-jar_2_1-1.dtd'>

<sun-ejb-jar>
  <enterprise-beans>
    <ejb>
      <ejb-name>ReportEJB </ejb-name>
      <jndi-name>ReportEJB </jndi-name>
    </ejb>
  </enterprise-beans>
</sun-ejb-jar>
```

sun-application.xml

Code Example B-23 sun-application.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE sun-application PUBLIC "-//Sun Microsystems, Inc.//DTD Application
Server 8.1 J2EE Application 1.4//EN"
'http://www.sun.com/software/appserver/dtds/sun-application_1_4-0.dtd'>
<sun-application>
  <security-role-mapping>
    <role-name>MANAGER-ROLE</role-name>
    <group-name>manager</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>EMPLOYEE-ROLE</role-name>
    <group-name>employee</group-name>
  </security-role-mapping>
</sun-application>
```

Troubleshooting the Agent Deployment

This appendix explains how you can resolve problems that you may encounter while deploying or using the J2EE Policy Agents.

Be sure to also check the Policy Agents Release Notes at http://docs.sun.com/db/coll/S1_IdServPolicyAgent_21 to see if the problem that you encounter is a known limitation of the agent. If workarounds are available for such problems, they will be provided in the release notes.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
1.	The agent installation program hangs and does not display the welcome message or any other message.	A previous run of the agent installation program may have failed or was terminated by the user and the lock file generated by it is still present in your system.	Search for the lock file and delete it. The lock file <code>productregistry.access.lock</code> or <code>productregistry.access.tmp</code> can be found in the location where the <code>productregistry</code> file is present. On Solaris, it is under the directory <code>/var/sadm/install</code> . On Windows 2000 systems, it is present under the directory <code>C:\WINNT\system32</code> .
2.	After installing the agent, the Identity Server fails to startup or does not allow any user to login.	The agent has been installed on the application server instance on which Identity Server is installed.	This is not a supported configuration. You can instead install the agent on a different instance of application server on the same machine, if necessary.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
3.	The agent installation program cannot proceed due to a previous installation. However, the previous installation was uninstalled before starting the new installation.	The previous uninstallation of the agent failed to remove the necessary entries from the productregistry file and/or left behind the Solaris package due to errors.	<p>You must search and cleanup the productregistry file. See the solution for symptom number 1 for the location of the file. This is a simple XML file that contains information regarding various Sun products and should be carefully edited to remove the references of any of the agent components. The agent components may be identified by the name of the agent that is being installed. If no other Sun product has been installed on your system, you can safely delete this file.</p> <p>If this is observed on a Solaris system, you must also uninstall the Solaris package corresponding to the agent installation. To identify the agent package, you can use a command such as:</p> <pre># pkginfo grep Agent</pre>
4.	The agent does not require users to login before access is granted to the application.	The application has not been configured to use the agent.	Refer to the section Post-Installation Tasks for steps to enable the agent for a protected application.
		The resources match entries in the Not-enforced List.	Make sure that the resources being accessed do not match the entries in Not-enforced List, and ensure that the list is not empty and inverted.
		The agent filter mode is set to NONE.	Change the agent filter mode to ALL or J2EE_POLICY as necessary.
5.	Agent denies access to all requests.	The agent and Identity Server have been installed on the same machine and the browser may not be setting the HOST header correctly when redirected from Identity Server to the agent.	Enable Port Check functionality. See the section Enabling Port Check Functionality .
		The application server is running as a user who does not have write privileges to the audit log directory of the agent.	Refer to the path specified in the <code>AMAgent.properties</code> file for the agent's local audit file and grant the necessary write permissions for the user of the application server process.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		The agent filter is configured for a mode that enforces URL policies and no applicable URL policies have yet been defined in Identity Server.	Define the appropriate URL policies in Identity Server.
		The agent filter is configured for a mode that enforces URL policies and the system time on the agent machine is not in sync with the system time on the Identity Server machine.	Synchronize the time of the agent machine with the time on the Identity Server machine. See the section Synchronizing the Agent Host and the Sun ONE Identity Server 6.1 Host for details.
		The agent filter is configured for a mode that does not support J2EE Policies and the resources being accessed are protected by declarative security constraints.	Change the agent filter mode to a mode that supports J2EE policy such as ALL or J2EE_POLICY.
		The agent filter is configured for a mode that supports J2EE policies but they are being negatively evaluated by the agent.	Refer to the solution for symptom number 6.
		The agent is unable to validate user's session token issued by Identity Server.	Make sure that the agent is installed on the same domain that is specified as the cookie domain in Identity Server. If not, enable CDSSO functionality. If that is not the case, try changing the value of the property <code>com.sun.am.policy.amFilter.sotoken.urldecode</code>
		The agent is configured for CDSSO and the validity time of the authorization response is smaller than the processing time required by the agent.	Set an appropriate value for the property <code>com.sun.am.policy.amFilter.cdssoclock.skew</code>
		The Login URL specified in the file <code>AMAgent.properties</code> is not reachable by the agent.	Make sure that the Identity Server Login URL is reachable from the machine where the agent is installed.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		The application server uses JDK version below 1.4 and no JCE/JSSE was selected during agent installation for the JDK used by application server.	Reinstall the agent and select the appropriate checkboxes as necessary to install JCE and/or JSSE for the JDK used by the application server.
		The Identity Server is installed with SSL and the agent cannot communicate with it correctly.	Refer to the solution for symptom number 9.
6.	The agent fails to evaluate J2EE declarative security policies or J2EE programmatic security APIs for the protected applications.	The protected application does not have the agent filter installed.	Redeploy the application with the agent filter installed and the login-configuration added as described in the Chapters 2 and 3 of this guide.
		The agent filter is operating in a mode that does not support J2EE policies.	Change the agent filter mode to either ALL or J2EE_POLICY. Change the filter mode to J2EE_POLICY
		The agent realm is not installed for the application server.	Refer to the section Post-Installation Tasks for the agent to ensure that the agent realm is properly configured.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		<p>The agent realm is installed but not correctly configured.</p>	<p>Make sure that the following properties are specified correctly in the <code>AMAgent.properties</code> file:</p> <pre>com.sun.am.policy.amRealm.organization.dn com.sun.am.policy.amRealm.peoplecontainer.level com.iplanet.am.directory.host com.iplanet.am.directory.ssl.enabled com.iplanet.am.directory.port</pre> <p>Also make sure that the DirDN values for <code>User1(cn=puser)</code> and <code>User2(cn=dsameuser)</code> specified in the <code>serverconfig.xml</code> file present under the directory <code>agent_config_directory/ums</code> have valid values corresponding to the installation organization and root-suffix.</p>
		<p>The agent realm is installed and configured correctly but has been disabled previously.</p>	<p>Enable the agent realm. Refer to the section Disabling the Agent Realm for steps.</p>
		<p>No password was specified for <code>amAdmin</code> user during the agent installation.</p>	<p>Encrypt the original password for <code>amAdmin</code> user as specified during the installation of Identity Server using the <code>agentadmin</code> tool. Enter the encrypted password in the <code>DirPassword</code> elements in <code>serverconfig.xml</code> file found under the directory <code>agent_config_directory/ums</code>. This password must be entered in two distinct places in this file for <code>User1(cn=puser)</code> and <code>User2(cn=dsameuser)</code>.</p>

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		<p>The password specified for amAdmin user during the agent installation is different from the amAdmin user password specified during the initial installation of Identity Server.</p>	<p>The amAdmin user password required by the agent is the original amAdmin user password that was created when Identity Server was installed. If subsequently the password for amAdmin user was changed, the old password should be used for installing the agent and not the new one. Take the old password and encrypt it using the agentadmin tool. Enter the encrypted password in the DirPassword elements in serverconfig.xml file found under <i>agent_config_directory</i>/ums. This password must be entered in two distinct places in this file for User1(cn=puser) and User2(cn=dsameuser).</p>
		<p>The specified role-to-principal mapping is incorrect.</p>	<p>Make sure that the specified role-to-principal mapping for the protected application is correct and maps to actual users and/or roles as they exist in Identity Server.</p>
		<p>The Identity Server's Directory Server is not available.</p>	<p>Make sure that the Identity Server's Directory Server is available and reachable from the machine on which the agent is installed.</p>
7.	<p>Accessing a protected resource results in HTTP 404 not found error.</p>	<p>The agent and Identity Server have been installed on the same machine and the browser being used may not be setting the HOST header correctly when redirected from Identity Server to the agent.</p>	<p>Enable Port Check Functionality. See the section Enabling Port Check Functionality.</p>
		<p>The resource is protected by a J2EE declarative security constraint which is not being evaluated correctly and the server is trying to display the form-error-page that does not exist within the application.</p>	<p>Make sure that the resource specified by form-error-page in the web application's web.xml deployment descriptor actually exists within the application.</p>

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		The resource is being accessed by a legacy browser such as Netscape 4.x and during the installation of the agent no valid value was entered for primary application context path field.	Reinstall the agent and specify a valid primary application context path value.
		The agent is operating in the CDSSO mode and no valid value was entered for the primary application context path field during agent installation.	Reinstall the agent and specify a valid primary application context path.
8.	On Solaris 8 platform, the agent for application server that uses JDK 1.4 hangs when Identity Server or Directory Server are configured for SSL.	The agent uses the JSSE shipped with the application server's JDK (1.4) to communicate with the Identity Server services. This implementation of JSSE requires the use of a secure random device (/dev/random) that is not included in Solaris 8 distribution.	In order to work around this issue you can either comment out the line <code>securerandom.source=file:/dev/random</code> in the <code>\$JAVA_HOME/lib/security/java.security</code> file, or install the OS patch 112438-02.
9.	When Identity Server is SSL Enabled, the agent denies access to all resources.	The agent does not have the root CA certificate of the signer of the certificate used by Identity Server.	Install the appropriate root CA certificate in the keystore used by the application server on which the agent is installed.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		<p>In the case of IBM WebSphere 5.0/5.1 agent, the sample URL provider for HTTPS protocol that was installed as part of the samples deployed for WebSphere 5.0/5.1 conflicts with the JSSE implementation used by the agent.</p>	<p>Remove the sample URL provider for HTTPS protocol using the following steps:</p> <ol style="list-style-type: none"> 1. Logon to the WebSphere Application Server Administration Console. 2. In the left hand panel, click on the Resources node. This will expand the node to reveal various resources that can be configured for the Server. 3. Click on the URL Providers resource link. This will load the corresponding properties on the right side content panel. 4. In the right side content panel, make sure that the Scope node is expanded. If not, click on this node to expand it. 5. Select the radio button next to the Server entry under the Scope node and click on the Apply button. 6. In the list of properties displayed in the lower portion of this panel, if you see Samples URL Provider - https entry, then this provider must be removed. Select the check box next to this provider and click on the Delete button. 7. Click on the Save link at the top of the page to save this change to the master configuration. This will bring you to a page that has the Save button. Click on this Save button to commit your changes. 8. Restart the WebSphere Application Server for the changes to take effect.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
10.	The agent is unable to fetch the LDAP attributes for the logged on user.	The requested resource that expects to use the attributes is in the not-enforced list of the agent.	Change the not-enforced list so that the requested resource is enforced. The agent does not provide support for LDAP attributes for resources that are not-enforced.
		The agent is not configured to communicate with the Identity Server's directory server correctly.	<p>Make sure that the following properties are set correctly in the <code>AMAgent.properties</code> file:</p> <pre>com.sun.am.policy.amRealm.organization.dn com.sun.am.policy.amRealm.peoplecontainer.level com.iplanet.am.directory.host com.iplanet.am.directory.ssl.enabled com.iplanet.am.directory.port</pre> <p>Also make sure that the DirDN values for <code>User1(cn=puser)</code> and <code>User2(cn=dsameuser)</code> specified in the <code>serverconfig.xml</code> file present under the directory <code>agent_config_directory/ums</code> have valid values corresponding to the installation organization and root-suffix.</p>
		No password for <code>amAdmin</code> user was specified during the agent installation.	Encrypt the original password for the <code>amAdmin</code> user as specified during the installation of Identity Server using the <code>agentadmin</code> tool. Enter the encrypted password in the <code>DirPassword</code> elements in <code>serverconfig.xml</code> found under the directory <code>agent_config_directory/ums</code> . This password must be entered in two distinct places in this file for <code>User1(cn=puser)</code> and <code>User2(cn=dsameuser)</code> .

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
		The password specified for amAdmin user during the agent installation is different from the amAdmin user password specified during the initial installation of Identity Server.	The amAdmin user password required by the agent is the original amAdmin user password that was created when Identity Server was installed. If subsequently the password for amAdmin user was changed, the old password should be used for installing the agent and not the new one. Take the old password and encrypt it using the agentadmin tool. Enter the encrypted password in the DirPassword elements in serverconfig.xml file found under the directory agent_config_directory/ums. This password must be entered in two distinct places in this file for User1(cn=puser) and User2(cn=dsameuser).
		The Identity Server's Directory Server is not available.	Make sure that the Identity Server's Directory Server is available and reachable from the machine on which the agent is installed.
11.	When some user information is changed in Identity Server, the changes do not reflect in the WebLogic Server protected by the agent, unless it is restarted.	The startup scripts for the agent protected server were modified such that the agent's classpath appears after the WebLogic Server's classpath.	Edit the startup scripts of the agent protected server and place the agent's classpath before the WebLogic server's classpath.
12.	The Identity Server logs indicate that it is unable to send notifications to the application server protected by the agent.	The agent is installed on a server with preferred listening protocol set to HTTPS and the root CA certificate for the signer of agent's server certificate is not available in the keystore used by Identity Server.	Add the root CA certificate for the signer of agent's server certificate to the keystore used by Identity Server.
		No valid value was entered for the primary application context path during agent installation.	Reinstall the agent and specify a valid primary application context path.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
13.	When accessing a resource protected by the Sun ONE Application Server 7.0 agent, the browser displays a blank page with the URL pointing to <code>j_security_check</code> .	This can happen if the entire web application is protected by declarative security constraints in <code>web.xml</code> deployment descriptor.	Edit the <code>AMAgent.properties</code> file and add a value corresponding to the protected application for the property: <code>com.sun.am.policy.amFilter.login.referer.map</code> The value of this property should be a URI that will be used to send the user once the form-login sequence has completed successfully.
14.	When accessing a servlet/JSP protected by the Sun ONE Application Server 7.0 agent that in turn invokes other servlets/JSPs, the agent is unable to enforce URL policies on subsequent resources.	This is a limitation of the implementation of Servlet Filters in Sun ONE Application Server 7.0	Use J2EE declarative security to protect such resources.

Table C-1 Troubleshooting Instructions

	Symptom	Possible Cause	Solution
15.	<p>The following warning messages appear on the console window from which the application server is started or in the application server logs:</p> <p>Bad level value for property: com.ipplanet.services.debug.level</p> <p>Bad level value for property: com.sun.identity.agents.logging.level</p> <p>Bad level value for property: com.sun.am.policy.amFilter.audit.level</p>	<p>These messages are logged by the JDK 1.4 logging framework in use, which treats the <code>AMAgent.properties</code> file as the logging configuration file.</p>	<p>These messages are harmless and can be safely ignored. If you wish to remove these messages, you should create a new properties file (for example <code>agentlog.properties</code>) and copy the following configuration properties from the file <code>AMAgent.properties</code> into this new file:</p> <pre>com.ipplanet.am.logstatus=ACTIVE ipplanet-am-logging-remote-handler=com.sun.identity.log.handlers.RemoteHandler ipplanet-am-logging-remote-formatter=com.sun.identity.log.handlers.RemoteFormatter ipplanet-am-logging-remote-buffer-size=1</pre> <p>Once this new file is created, you should set the value of the system property <code>java.util.logging.config.file</code> for the application server to the complete path of this new file.</p> <p>Depending upon the application server for which you have installed the agent, you may have to edit either the startup script of the application server, or the configuration properties/xml file where such system properties can be defined for the application server.</p>

Index

A

- agent
 - tools
 - agentadmin script 275
 - unconfiguring 290
- Agent API 299
- agent authentication provider
 - configuring for BEA WebLogic Server 7.0/8.1 117
- agent configuration file 154
- agent filter
 - installing
 - on Apache Tomcat Server 129
 - on BEA WebLogic 8.1 SP2/SP3 Portal 145
 - on BEA WebLogic Server 6.1 SP2 112
 - on BEA WebLogic Server 7.0/8.1 118
 - on IBM WebSphere Application Server 5.0/5.1 115
 - on Macromedia JRun 4 136
 - on Oracle 9iAS R2 and Oracle 10g 131
 - on PeopleSoft 8.3/8.4/8.8 122
- modes 156
 - ALL 159
 - J2EE_POLICY 158
 - NONE 157
 - SSO_ONLY 157
 - URL_POLICY 158
- agent host with Identity Server host 152
- agent properties
 - list of properties 179
- agent realm
 - disabling 159
 - for BEA WebLogic Server Agent 160
 - for IBM WebSphere Application Server Agent 160
 - for Oracle9iAS and Oracle 10g Agent 161
 - for Sun Java System Application Server 8.1 Agent 162
 - for Sun ONE Application Server 7.0 Agent 159
- removing
 - BEA WebLogic Server 6.1 SP2 Agent 302
- agent tools
 - to configure application server instance 277
 - to encrypt strings 276
- agentadmin
 - agent tools 275
- agentauthenticatorservlet
 - configuring 124
- agents
 - See policy agents
- amAdmin
 - password 86
- amAdmin user
 - password 54
- AMAgent
 - properties
 - reference 179
- amldapuser
 - password 54, 86
- Apache Tomcat Server 4.1.27 agent
 - assembly descriptors
 - web.xml 359
 - pre-installation tasks 39
 - supported servers 25

- application configuration
 - for WebLogic 7.0 SP1 agent [118](#)
- application.xml
 - standard deployment descriptors [350](#)
- assembly descriptors
 - Apache Tomcat Server 4.1.27 [359](#)
 - BEA WebLogic 6.1SP2 [352](#)
 - BEA WebLogic Server 7.0 SP2 and BEA WebLogic Server 8.1 [357](#)
 - IBM WebSphere Application Server 5.0/5.1 [354](#)
 - Macromedia JRun 4 [360](#)
 - Oracle 9iAS R2 and Oracle 10g [360](#)
 - Sun Java System Application Server 8.1 [362](#)
 - Sun ONE Application Server 7.0 [351](#)

B

- BEA WebLogic 6.1 SP2 agent
 - assembly descriptors
 - weblogic.xml [353](#)
 - weblogic-ejb-jar.xml [353](#)
 - supported servers [24](#)
- BEA WebLogic 7.0 SP2 and BEA WebLogic 8.1 agent
 - assembly descriptors
 - weblogic.xml [358](#)
 - weblogic-ejb-jar.xml [358](#)
 - supported servers
 - BEA WebLogic Server 7.0 SP2 [24](#)
 - BEA WebLogic Server 8.1 [25](#)
- BEA WebLogic 8.1 SP2/SP3 Server/Portal agent
 - supported servers [25](#)

C

- caching realm
 - creating [111](#)
- CDSSO
 - enabling [151](#)
- Class AmFilterManager [299](#)
- Class AmSSOCache [300](#)
- config reload interval [56](#)

- configuring
 - agent (for an additional application server instance)
 - for Apache Tomcat Server 4.1.27 [285](#)
 - for BEA WebLogic (all versions) [280](#)
 - for IBM WebSphere Application Server 5.0/5.1 [280](#)
 - for Macromedia JRun 4 [285](#)
 - for Oracle 9iAS R2 and Oracle 10g [287](#)
 - for PeopleSoft 8.3/8.4/8.8 [283](#)
 - for SAP Enterprise Portal 6.0 SP2 [287](#)
 - for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1 [288](#)
 - for Sun Java System Application Server 8.1 [288](#)
 - for Sun ONE Application Server 7.0 [279](#)

- context path
 - primary application [57](#)

- cookie reset [177](#)

- creating
 - PeopleSoft 8.3/8.4/8.8
 - DEFAULT_USER [35](#)

- custom realm
 - creating [110](#)

D

- DEFAULT_USER
 - PeopleSoft 8.3/8.4/8.8 agent
 - creating [35](#)

- descriptors
 - assembly [350](#)
 - standard deployment [345](#)

- developer information [12](#)

- differences
 - J2EE agents and web agents [27](#)

- disabling
 - agent realm [159](#)
 - for BEA WebLogic Server Agent [160](#)
 - for IBM WebSphere Application Server Agent [160](#)
 - for Oracle9iAS and Oracle 10g Agent [161](#)
 - for Sun Java System Application Server 8.1 Agent [162](#)

- for Sun ONE Application Server 7.0 Agent [159](#)
- downloads
 - Sun ONE software [12](#)

E

- ejb-jar.xml
 - standard deployment descriptors [349](#)
- enabling security
 - IBM WebSphere Application Server 5.0/5.1 [114](#)
- Enterprise Manager
 - configuring the agent [133](#)
- Enterprise Manager Console
 - installing the application [133](#)

F

- failover
 - enabling [170](#)
- file realm
 - configuring [112](#)
- form login
 - customizing the agent response [169](#)
- FQDN handling
 - configuring [176](#)

G

- generating
 - state file [339](#)

H

- hot-swap mechanism [154](#)

- IBM WebSphere Application Server 5.0/5.1 agent
 - assembly descriptors
 - ibm-application-bnd.xmi [356](#)
 - ibm-application-ext.xmi [357](#)
 - ibm-ejb-jar-bnd.xmi [355](#)
 - ibm-ejb-jar-ext.xmi [356](#)
 - ibm-web.bnd.xmi [354](#)
 - ibm-web-ext.xmi [355](#)
 - supported servers
 - IBM WebSphere Application Server 5.0 [24](#)
 - IBM WebSphere Application Server 5.1 [25](#)
- ibm-application-bnd.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [356](#)
- ibm-application-ext.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [357](#)
- ibm-ejb-jar-bnd.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [355](#)
- ibm-ejb-jar-ext.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [356](#)
- ibm-web-bnd.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [354](#)
- ibm-web-ext.xmi
 - IBM WebSphere 5.0/5.1
 - assembly descriptors [355](#)
- installation
 - silent [339](#)
 - using state file [340](#)
- installation program
 - command line-based [83](#)
 - GUI-based [51](#)
 - launching [46](#)
 - launching in the Command Line Mode [48](#)
 - launching in the GUI Mode [47](#)
 - launching on Solaris and HP-UX [46](#)
 - launching on Windows 2000 [49](#)
 - using [51](#)

J

- J2EE agents
 - differences with web agents 27

L

- LDAP attributes
 - as cookies 175
 - as HTTP headers 174
 - as request attributes 175
 - enabling 174
- list constructs
 - in AMAgent.properties 154
- login attempt limit 171

M

- Macromedia JRun 4 agent
 - assembly descriptors 360
 - supported servers 25
- map constructs
 - in AMAgent.properties 155

N

- not-enforced list
 - inverting 173
- not-enforced list cache
 - enable 57

O

- Oracle 9iAS R2 and Oracle 10g agent
 - assembly descriptors
 - orion-application.xml 361
 - orion-ejb-jar.xml 361
 - orion-web.xml 361

- pre-installation 40
- supported servers
 - Oracle 9i Application Server Release 2 25
 - Oracle Application Server 10g 25
- orion-application.xml
 - Oracle 9iAS R2 and Oracle 10g
 - assembly descriptors 361
- orion-ejb-jar.xml
 - Oracle 9iAS R2 and Oracle 10g
 - assembly descriptors 361
- orion-web.xml
 - Oracle 9iAS R2 and Oracle 10g
 - assembly descriptors 361

P

- PeopleSoft 8.3/8.4/8.8 agent
 - creating DEFAULT_USER 35
 - PeopleCode
 - installing 36
 - registering 37
 - supported servers 24
- policy agents
 - component
 - agent filter 20
 - agent realm 20
 - configuration 153
 - configuring for an additional application server
 - instance 277
 - general usage examples 14
 - how they work 20
 - launching the installation program 46
 - launching the uninstallation program 308
 - post-installation tasks 107
 - post-uninstallation tasks 324
 - pre-installation 31
 - pre-uninstallation tasks 301
 - specialized agent usage examples 17
 - supported servers 24
 - unconfiguring for an application server
 - instance 290
 - uses 14
 - using the installation program 51
 - using the uninstallation program 312

- port check functionality
 - enabling 178
- post-installation
 - tasks 107
- pre-installation tasks 31
- pre-uninstallation
 - tasks 301
- primary application
 - context path 57
- professional services 12
- properties
 - AMAgent 179

R

- redirect attempt limit 172
- role-to-principal mappings
 - creating 120

S

- SAP Enterprise Portal 6.0 SP2 agent
 - disabling
 - SAP* User 41
 - pre-installation 41
 - supported servers 25
 - user-mapping scheme 41
- SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1 agent
 - supported servers 25
- silent
 - installation 339
 - uninstallation 339
- solaris
 - patches 12
 - support 12
- standard deployment descriptors
 - application.xml 350
 - ejb-jar.xml 349
 - web.xml 346
- state file

- for installation 340
 - for uninstallation 342
- generating 339
 - uninstallation 341
- Sun Java System Application Server 8.1 agent
 - assembly descriptors
 - sun-application.xml 364
 - sun-ejb-jar.xml 363
 - sun-web.xml 363
 - supported servers 25
- Sun ONE
 - support 12
- Sun ONE Application Server 7.0 agent
 - assembly descriptors
 - sun-application.xml 352
 - sun-ejb-jar.xml 352
 - sun-web.xml 351
 - supported servers 24
- Sun ONE Identity Server
 - related information 12
- sun-application.xml
 - Sun Java System Application Server 8.1
 - assembly descriptors 364
 - Sun ONE Application Server 7.0
 - assembly descriptors 352
- sun-ejb-jar.xml
 - Sun Java System Application Server 8.1
 - assembly descriptors 363
 - Sun ONE Application Server 7.0
 - assembly descriptors 352
- sun-web.xml
 - Sun Java System Application Server 8.1
 - assembly descriptors 363
 - Sun ONE Application Server 7.0
 - assembly descriptors 351
- support
 - professional services 12
 - solaris 12
 - Sun ONE 12
- synchronizing 152

T

- troubleshooting

instructions [365](#)

U

Unconfiguring

- agent (for an application server instance)
 - for Apache Tomcat Server 4.1.27 [294](#)
 - for BEA WebLogic Server (All Versions) [292](#)
 - for IBM WebSphere Application Server [292](#)
 - for Macromedia JRun 4 [295](#)
 - for Oracle 9iAS R2 and Oracle 10g Server Instance [296](#)
 - for PeopleSoft 8.3/8.4/8.8 [293](#)
 - for SAP Enterprise Portal 6.0 SP2 [296](#)
 - for SAP Enterprise Portal 6.0 SP2 and Web Application Server 6.20 SP1 [297](#)
 - for Sun Java System Application Server 8.1 [298](#)
 - for Sun ONE Application Server 7.0 [291](#)

uninstallation

- program
 - command line-based [318](#)
 - GUI-based [312](#)
 - launching [308](#)
 - launching on Solaris, HP-UX, AIX and Linux [308](#)
 - launching on Windows 2000 [311](#)
 - using [312](#)
- silent [339](#)
- using state file [341](#), [342](#)

URL policies

- creating for PeopleSoft 8.3/8.4/8.8 [124](#)
- creating for SAP Enterprise Portal 6.0 agent [135](#)

user mapping mode

- configuring for SAP Enterprise Portal 6.0 SP2 agent [135](#)
- configuring for SAP Enterprise Portal 6.0 SP2 agent and Web Application Server 6.20 SP1 [141](#)

W

web agents

- differences with J2EE agents [27](#)

web.xml

- Apache Tomcat Server 4.1.27
 - assembly descriptors [359](#)
 - standard deployment descriptor [346](#)

weblogic.xml

- BEA WebLogic 6.1 SP2
 - assembly descriptors [353](#)
- BEA WebLogic 7.0 SP2 and BEA WebLogic 8.1
 - assembly descriptors [358](#)

weblogic-ejb-jar.xml

- BEA WebLogic 6.1 SP2
 - assembly descriptors [353](#)
- BEA WebLogic 7.0 SP2 and BEA WebLogic 8.1
 - assembly descriptors [358](#)

web-tier declarative security

- enabling [163](#)

web-tier security

- details [168](#)