Before using this information and the product it supports, read the information in Appendix C, “Notices,” on page 61.
# Contents

**Preface** ........................................... v
Who should read this book .................................. v
What this book contains .................................. v
Publications ........................................... vi
  - Release information ................................ vi
  - Base information ................................ vi
  - Web security information ........................... vi
  - Developer references ............................... vii
  - Technical supplements ............................. viii
  - Related publications .............................. viii
  - Accessing publications online .................... xi
Accessibility .......................................... xi
Contacting software support .............................. xi
Conventions used in this book ............................ xii
  - Typeface conventions .............................. xii
  - Operating system differences ..................... xii

**Chapter 1. Introduction and overview** ................. 1
The Tivoli Access Manager security model .................. 1
Integrating Tivoli Access Manager and WebLogic Server ... 2
  - Tivoli Access Manager Security Service Provider Interface components .......................... 2
  - Policy and role deployment ........................ 4
  - Resources and roles ................................ 4
  - Using Tivoli Access Manager authentication .... 5
Logging and auditing .................................. 7
Reliability, Availability, and Scalability .................... 7

**Chapter 2. Installation instructions** ................. 9
Supported platforms .................................... 9
Disk and memory requirements ........................... 9
Software prerequisites ................................ 10
  - Tivoli Access Manager policy server ............. 10
  - Tivoli Access Manager Authorization Server .... 10
  - Tivoli Access Manager WebSEAL or Tivoli Access Manager Plug-in for Web Servers ............. 10
  - BEA WebLogic Server ............................. 11
  - Tivoli Access Manager Java runtime .............. 11
Installing using the installation wizard .................. 11
install_amwl's options ................................ 13
Installing using native utilities ....................... 14
  - Installing on AIX .................................. 14
  - Installing on HP-UX ............................... 15
  - Installing on Solaris ............................. 15
  - Installing on Windows ............................. 16

**Chapter 3. Configuration procedures** ................. 19
Part 1: Configuring the Tivoli Access Manager Java runtime environment .................. 19
Part 2: Setting CLASSPATH for startWebLogic .................. 20
Part 3: Configuring Tivoli Access Manager for WebLogic .................. 21
  - Configuring Tivoli Access Manager for WebLogic using the Console Extension Web application .................. 21
  - Configuring Tivoli Access Manager for WebLogic from the command line .................. 23
Part 4: Configuring the Tivoli Access Manager Realm .................. 24
  - Configuring the Tivoli Access Manager Realm using the Console Extension Web application .................. 24
  - Configuring the Tivoli Access Manager Realm from the command line .................. 25
Part 5: Configuring for BEA WebLogic Server single signon .................. 26
  - Configuring single signon using a WebSEAL junction .................. 27
Configuring single signon using Tivoli Access Manager Plug-in for Web Servers ........................................... 28
Part 6: Configuring Tivoli Access Manager for WebLogic in BEA WebLogic Server multi-server environments including clustered environments ....................................................... 28
Part 7: Testing the configuration .................................................................................................................. 29

Chapter 4. Enabling Single Signon. ........................................................................................................... 31
Single sign-on with Tivoli Access Manager WebSEAL .................................................................................. 31

Chapter 5. Administration tasks .............................................................................................................. 33
Enabling the entitlements service on a Tivoli Access Manager Authorization server .................................. 33
User and group management with Tivoli Access Manager for WebLogic .................................................. 34
Using the demonstration application .......................................................................................................... 35
Usage tips .................................................................................................................................................... 36
Three strikes logon policy ............................................................................................................................... 37
Deleting the Tivoli Access Manager Realm ................................................................................................. 38
Unconfiguring Tivoli Access Manager for WebLogic ................................................................................... 39
Troubleshooting tips .................................................................................................................................. 39
  Single sign-on failure using forms-based login ......................................................................................... 39
  WebLogic Server throws memory exception ........................................................................................... 39
Limitations .................................................................................................................................................... 40
Known Issues and Work-arounds .................................................................................................................. 40

Chapter 6. Removal instructions. ............................................................................................................. 41
Removing from Solaris .................................................................................................................................. 41
Removing from Windows ................................................................................................................................. 41
Removing from AIX ...................................................................................................................................... 42
Removing from HP-UX ................................................................................................................................. 42

Appendix A. Properties Files Reference .................................................................................................. 45
amsspi.properties ....................................................................................................................................... 45
rbpf.properties ................................................................................................................................ .......... 47
amwlsljlog.properties ................................................................................................................................. 50

Appendix B. Command quick reference .................................................................................................. 53
AMWLSConfigure –action config ................................................................................................................ 54
AMWLSConfigure –action unconfig ............................................................................................................ 56
AMWLSConfigure –action create_realm ..................................................................................................... 57
AMWLSConfigure –action delete_realm ..................................................................................................... 59

Appendix C. Notices ............................................................................................................................... 61
Trademarks .................................................................................................................................................. 62

Glossary ....................................................................................................................................................... 65

Index ............................................................................................................................................................ 71
Preface

Welcome to IBM® Tivoli® Access Manager for BEA® WebLogic Server® (hereafter referred to as Tivoli Access Manager for WebLogic). This product extends IBM Tivoli Access Manager to support applications written for BEA WebLogic Server.

IBM® Tivoli® Access Manager (Tivoli Access Manager) is the base software that is required to run applications in the IBM Tivoli Access Manager product suite. It enables the integration of IBM Tivoli Access Manager applications that provide a wide range of authorization and management solutions. Sold as an integrated solution, these products provide an access control management solution that centralizes network and application security policy for e-business applications.

Note: IBM Tivoli Access Manager is the new name of the previously released software entitled Tivoli SecureWay® Policy Director. Also, for users familiar with the Tivoli SecureWay Policy Director software and documentation, the management server is now referred to as the policy server.


Who should read this book

The target audience for this administration guide includes:
- Security administrators
- Network system administrators
- IT architects

Readers should be familiar with:
- Internet protocols, including HTTP, TCP/IP, file transfer protocol (FTP), and Telnet
- Deployment and management of WebLogic Server systems
- Security management, including authentication and authorization

If you are enabling Secure Sockets Layer (SSL) communication, you also should be familiar with SSL protocol, key exchange (public and private), digital signatures, cryptographic algorithms, and certificate authorities.

What this book contains

This document contains the following chapters:
- Chapter 1, “Introduction and overview”
  Presents an overview of the authentication and authorization services provided by Tivoli Access Manager for WebLogic.
- Chapter 2, “Installation instructions”
  Describes how to install Tivoli Access Manager for WebLogic.
- Chapter 3, “Configuration procedures”
  Describes how to configure Tivoli Access Manager for WebLogic.
• Chapter 4, “Administration tasks”
  Describes how to use the demonstration application, and provides usage tips, troubleshooting information, and limitations.
• Chapter 5, “Removal instructions”
  Describes how to remove Tivoli Access Manager for WebLogic.

Publications

Review the descriptions of the Tivoli Access Manager library, the prerequisite publications, and the related publications to determine which publications you might find helpful. After you determine the publications you need, refer to the instructions for accessing publications online.

Additional information about the IBM Tivoli Access Manager for e-business product itself can be found at:


The Tivoli Access Manager library is organized into the following categories:

• “Release information”
• “Base information”
• “Web security information”
• “Developer references” on page vii
• “Technical supplements” on page viii

Release information

• IBM Tivoli Access Manager for e-business Read This First (GI11-4155-00)
  Provides information for installing and getting started using Tivoli Access Manager.
• IBM Tivoli Access Manager for e-business Release Notes (GI11-4156-00)
  Provides late-breaking information, such as software limitations, workarounds, and documentation updates.

Base information

• IBM Tivoli Access Manager Base Installation Guide (SC32-1362-00)
  Explains how to install and configure the Tivoli Access Manager base software, including the Web Portal Manager interface. This book is a subset of IBM Tivoli Access Manager for e-business Web Security Installation Guide and is intended for use with other Tivoli Access Manager products, such as IBM Tivoli Access Manager for Business Integration and IBM Tivoli Access Manager for Operating Systems.
• IBM Tivoli Access Manager Base Administration Guide (SC32-1360-00)
  Describes the concepts and procedures for using Tivoli Access Manager services. Provides instructions for performing tasks from the Web Portal Manager interface and by using the pdadmin command.

Web security information

• IBM Tivoli Access Manager for e-business Web Security Installation Guide (SC32-1361-00)
Provides installation, configuration, and removal instructions for the Tivoli Access Manager base software as well as the Web Security components. This book is a superset of IBM Tivoli Access Manager Base Installation Guide.

- **IBM Tivoli Access Manager Upgrade Guide (SC32-1369-00)**
  Explains how to upgrade from Tivoli SecureWay Policy Director Version 3.8 or previous versions of Tivoli Access Manager to Tivoli Access Manager Version 5.1.

- **IBM Tivoli Access Manager for e-business WebSEAL Administration Guide (SC32-1359-00)**
  Provides background material, administrative procedures, and technical reference information for using WebSEAL to manage the resources of your secure Web domain.

- **IBM Tivoli Access Manager for e-business IBM WebSphere Application Server Integration Guide (SC32-1368-00)**
  Provides installation, removal, and administration instructions for integrating Tivoli Access Manager with IBM WebSphere® Application Server.

- **IBM Tivoli Access Manager for e-business IBM WebSphere Edge Server Integration Guide (SC32-1367-00)**
  Provides installation, removal, and administration instructions for integrating Tivoli Access Manager with the IBM WebSphere Edge Server application.

- **IBM Tivoli Access Manager for e-business Plug-in for Web Servers Integration Guide (SC32-1365-00)**
  Provides installation instructions, administration procedures, and technical reference information for securing your Web domain using the plug-in for Web servers.

- **IBM Tivoli Access Manager for e-business BEA WebLogic Server Integration Guide (SC32-1366-00)**
  Provides installation, removal, and administration instructions for integrating Tivoli Access Manager with BEA WebLogic Server.

- **IBM Tivoli Access Manager for e-business IBM Tivoli Identity Manager Provisioning Fast Start Guide (SC32-1364-00)**
  Provides an overview of the tasks related to integrating Tivoli Access Manager and Tivoli Identity Manager and explains how to use and install the Provisioning Fast Start collection.

### Developer references

- **IBM Tivoli Access Manager for e-business Authorization C API Developer Reference (SC32-1355-00)**
  Provides reference material that describes how to use the Tivoli Access Manager authorization C API and the Tivoli Access Manager service plug-in interface to add Tivoli Access Manager security to applications.

- **IBM Tivoli Access Manager for e-business Authorization Java Classes Developer Reference (SC32-1350-00)**
  Provides reference information for using the Java™ language implementation of the authorization API to enable an application to use Tivoli Access Manager security.

- **IBM Tivoli Access Manager for e-business Administration C API Developer Reference (SC32-1357-00)**
  Provides reference information about using the administration API to enable an application to perform Tivoli Access Manager administration tasks. This document describes the C implementation of the administration API.
• IBM Tivoli Access Manager for e-business Administration Java Classes Developer Reference (SC32-1356-00)
  Provides reference information for using the Java language implementation of the administration API to enable an application to perform Tivoli Access Manager administration tasks.
• IBM Tivoli Access Manager for e-business Web Security Developer Reference (SC32-1358-00)
  Provides administration and programming information for the cross-domain authentication service (CDAS), the cross-domain mapping framework (CDMF), and the password strength module.

Technical supplements
• IBM Tivoli Access Manager for e-business Command Reference (SC32-1354-00)
  Provides information about the command line utilities and scripts provided with Tivoli Access Manager.
• IBM Tivoli Access Manager Error Message Reference (SC32-1353-00)
  Provides explanations and recommended actions for the messages produced by Tivoli Access Manager.
• IBM Tivoli Access Manager for e-business Problem Determination Guide (SC32-1352-00)
  Provides problem determination information for Tivoli Access Manager.
• IBM Tivoli Access Manager for e-business Performance Tuning Guide (SC32-1351-00)
  Provides performance tuning information for an environment consisting of Tivoli Access Manager with the IBM Tivoli Directory server as the user registry.

Related publications
This section lists publications related to the Tivoli Access Manager library.

The Tivoli Software Library provides a variety of Tivoli publications such as white papers, datasheets, demonstrations, redbooks, and announcement letters. The Tivoli Software Library is available on the Web at: http://www.ibm.com/software/tivoli/library/

The Tivoli Software Glossary includes definitions for many of the technical terms related to Tivoli software. The Tivoli Software Glossary is available, in English only, from the Glossary link on the left side of the Tivoli Software Library Web page http://www.ibm.com/software/tivoli/library/

IBM Global Security Kit
Tivoli Access Manager provides data encryption through the use of the IBM Global Security Kit (GSKit) Version 7.0. GSKit is included on the IBM Tivoli Access Manager Base CD for your particular platform, as well as on the IBM Tivoli Access Manager Web Security CDs, the IBM Tivoli Access Manager Web Administration Interfaces CDs, and the IBM Tivoli Access Manager Directory Server CDs.

The GSKit package provides the iKeyman key management utility, gsk7ikm, which is used to create key databases, public-private key pairs, and certificate requests. The following document is available on the Tivoli Information Center Web site in the same section as the IBM Tivoli Access Manager product documentation:
• IBM Global Security Kit Secure Sockets Layer and iKeyman User’s Guide (SC32-1363-00)
Provides information for network or system security administrators who plan to enable SSL communication in their Tivoli Access Manager environment.

**IBM Tivoli Directory Server**
IBM Tivoli Directory Server, Version 5.2, is included on the **IBM Tivoli Access Manager Directory Server** CD for the desired operating system.

**Note:** IBM Tivoli Directory Server is the new name for the previously released software known as:
- IBM Directory Server (Version 4.1 and Version 5.1)
- IBM SecureWay Directory Server (Version 3.2.2)


Additional information about IBM Tivoli Directory Server can be found at:


**IBM DB2 Universal Database**
IBM DB2® Universal Database™ Enterprise Server Edition, Version 8.1 is provided on the **IBM Tivoli Access Manager Directory Server** CD and is installed with the IBM Tivoli Directory Server software. DB2 is required when using IBM Tivoli Directory Server, z/OS™, or OS/390® LDAP servers as the user registry for Tivoli Access Manager.

Additional information about DB2 can be found at:

http://www.ibm.com/software/data/db2/

**IBM WebSphere Application Server**
IBM WebSphere Application Server, Advanced Single Server Edition 5.0, is included on the **IBM Tivoli Access Manager Web Administration Interfaces** CD for the desired operating system. WebSphere Application Server enables the support of both the Web Portal Manager interface, which is used to administer Tivoli Access Manager, and the Web Administration Tool, which is used to administer IBM Tivoli Directory Server. IBM WebSphere Application Server Fix Pack 2 is also required by Tivoli Access Manager and is provided on the **IBM Tivoli Access Manager WebSphere Fix Pack** CD.

Additional information about IBM WebSphere Application Server can be found at:


**IBM Tivoli Access Manager for Business Integration**
IBM Tivoli Access Manager for Business Integration, available as a separately orderable product, provides a security solution for IBM MQSeries®, Version 5.2, and IBM WebSphere® MQ for Version 5.3 messages. IBM Tivoli Access Manager for Business Integration allows WebSphere MQSeries applications to send data with privacy and integrity by using keys associated with sending and receiving applications. Like WebSEAL and IBM Tivoli Access Manager for Operating Systems, IBM Tivoli Access Manager for Business Integration, is one of the resource managers that use the services of IBM Tivoli Access Manager.
Additional information about IBM Tivoli Access Manager for Business Integration can be found at:


The following documents associated with IBM Tivoli Access Manager for Business Integration Version 5.1 are available on the Tivoli Information Center Web site:

- IBM Tivoli Access Manager for Business Integration Administration Guide (SC23-4831-01)
- IBM Tivoli Access Manager for Business Integration Problem Determination Guide (GC23-1328-00)
- IBM Tivoli Access Manager for Business Integration Release Notes (GI11-0957-01)
- IBM Tivoli Access Manager for Business Integration Read This First (GI11-4202-00)

IBM Tivoli Access Manager for WebSphere Business Integration Brokers

IBM Tivoli Access Manager for WebSphere Business Integration Brokers, available as part of IBM Tivoli Access Manager for Business Integration, provides a security solution for WebSphere Business Integration Message Broker, Version 5.0 and WebSphere Business Integration Event Broker, Version 5.0. IBM Tivoli Access Manager for WebSphere Business Integration Brokers operates in conjunction with Tivoli Access Manager to secure JMS publish/subscribe applications by providing password and credentials-based authentication, centrally-defined authorization, and auditing services.

Additional information about IBM Tivoli Access Manager for WebSphere Integration Brokers can be found at:


The following documents associated with IBM Tivoli Access Manager for WebSphere Integration Brokers, Version 5.1 are available on the Tivoli Information Center Web site:

- IBM Tivoli Access Manager for WebSphere Business Integration Brokers Administration Guide (SC32-1347-00)
- IBM Tivoli Access Manager for WebSphere Business Integration Brokers Release Notes (GI11-4154-00)
- IBM Tivoli Access Manager for Business Integration Read This First (GI11-4202-00)

IBM Tivoli Access Manager for Operating Systems

IBM Tivoli Access Manager for Operating Systems, available as a separately orderable product, provides a layer of authorization policy enforcement on UNIX systems in addition to that provided by the native operating system. IBM Tivoli Access Manager for Operating Systems, like WebSEAL and IBM Tivoli Access Manager for Business Integration, is one of the resource managers that use the services of IBM Tivoli Access Manager.

Additional information about IBM Tivoli Access Manager for Operating Systems can be found at:


The following documents associated with IBM Tivoli Access Manager for Operating Systems Version 5.1 are available on the Tivoli Information Center Web site:
IBM Tivoli Access Manager for Operating Systems Installation Guide (SC23-4829-00)
IBM Tivoli Access Manager for Operating Systems Administration Guide (SC23-4827-00)
IBM Tivoli Access Manager for Operating Systems Problem Determination Guide (SC23-4828-00)
IBM Tivoli Access Manager for Operating Systems Release Notes (GI11-0951-00)
IBM Tivoli Access Manager for Operating Systems Read Me First (GI11-0949-00)

IBM Tivoli Identity Manager
IBM Tivoli Identity Manager Version 4.5, available as a separately orderable product, enables you to centrally manage users (such as user IDs and passwords) and provisioning (that is providing or revoking access to applications, resources, or operating systems.) Tivoli Identity Manager can be integrated with Tivoli Access Manager through the use of the Tivoli Access Manager Agent. Contact your IBM account representative for more information about purchasing the Agent.

Additional information about IBM Tivoli Identity Manager can be found at:


Accessing publications online
The publications for this product are available online in Portable Document Format (PDF) or Hypertext Markup Language (HTML) format, or both in the Tivoli software library: http://www.ibm.com/software/tivoli/library

To locate product publications in the library, click the Product manuals link on the left side of the library page. Then, locate and click the name of the product on the Tivoli software information center page.

Product publications include release notes, installation guides, user’s guides, administrator’s guides, and developer’s references.

**Note:** To ensure proper printing of PDF publications, select the Fit to page check box in the Adobe Acrobat Print window (which is available when you click File + Print).

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You also can use the keyboard instead of the mouse to operate all features of the graphical user interface.

Contacting software support

Before contacting IBM Tivoli Software Support with a problem, refer to the IBM Tivoli Software Support site by clicking the Tivoli support link at the following Web site: http://www.ibm.com/software/support/

If you need additional help, contact software support by using the methods described in the IBM Software Support Guide at the following Web site:
http://techsupport.services.ibm.com/guides/handbook.html

The guide provides the following information:
Registration and eligibility requirements for receiving support
Telephone numbers, depending on the country in which you are located
A list of information you should gather before contacting customer support

Conventions used in this book

This reference uses several conventions for special terms and actions and for operating system-dependent commands and paths.

Typeface conventions

The following typeface conventions are used in this reference:

**Bold**
Lowercase commands or mixed case commands that are difficult to distinguish from surrounding text, keywords, parameters, options, names of Java classes, and objects are in bold.

*Italic*
Variables, titles of publications, and special words or phrases that are emphasized are in italic.

Monospace

Code examples, command lines, screen output, file and directory names that are difficult to distinguish from surrounding text, system messages, text that the user must type, and values for arguments or command options are in monospace.

Operating system differences

This book uses the UNIX convention for specifying environment variables and for directory notation. When using the Windows command line, replace $variable with %variable% for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. If you are using the bash shell on a Windows system, you can use the UNIX conventions.
Chapter 1. Introduction and overview

Tivoli Access Manager for WebLogic is an extension to Tivoli Access Manager that secures access to BEA WebLogic Server applications using the security features of Tivoli Access Manager. Using the BEA WebLogic Server Security Service Provider Interface, Tivoli Access Manager for WebLogic authenticates clients using a user registry administered by Tivoli Access Manager. IBM Tivoli Access Manager WebSEAL (WebSEAL) or IBM Tivoli Access Manager Plug-in for Web Servers can be used to extend the security features of Tivoli Access Manager for WebLogic to provide support for an end-user single sign-on.

Tivoli Access Manager for WebLogic enables WebLogic Server applications to use Tivoli Access Manager security without requiring any coding or deployment changes.

A Tivoli Access Manager secure domain must be deployed prior to installation of Tivoli Access Manager for WebLogic.

Users who are new to Tivoli Access Manager should review the Tivoli Access Manager security model before deploying a secure domain. A brief summary of the security model is presented here.

The Tivoli Access Manager security model

Tivoli Access Manager is a complete authorization and network security policy management solution that provides end-to-end protection of resources over geographically dispersed intranets and extranets.

Tivoli Access Manager features state-of-the-art security policy management. In addition, it supports authentication, authorization, data security, and resource management capabilities. You use Tivoli Access Manager in conjunction with standard Internet-based applications to build highly secure and well-managed intranets and extranets.

At its core, Tivoli Access Manager provides:

• An authentication framework
  Tivoli Access Manager supports a wide range of authentication mechanisms, including: certificates, Basic authentication, forms and HTTP headers.

• An authorization framework
  Tivoli Access Manager provides a framework for authorization policy management. Authorization policy is managed centrally and distributed automatically to access enforcement points across the enterprise. The Tivoli Access Manager authorization service provides permit and deny decisions on access requests for native Tivoli Access Manager servers and third-party applications.

WebSEAL is the Tivoli Access Manager resource security manager for Web-based resources. WebSEAL is a high performance, multi-threaded Web server that applies fine-grained security to protected Web resources.

Tivoli Access Manager Plug-in for Web Servers integrates with Tivoli Access Manager to provide a complete security solution for your Web resources. The
Integrating Tivoli Access Manager and WebLogic Server

Tivoli Access Manager for WebLogic, Version 5.1, supports:

- BEA WebLogic Server version 7.0 SP2
- BEA WebLogic Server version 8.1 SP1

Tivoli Access Manager for WebLogic Version 5.1 provides a full security framework for BEA WebLogic Server using the Security Service Provider Interface (SSPI).

**Note:** Tivoli Access Manager for WebLogic Version 5.1 does not support the BEA WebLogic Server custom realm. Support for the BEA WebLogic Server custom realm was part of Tivoli Access Manager for WebLogic Version 4.1.

BEA WebLogic Server provides SSPI for third-party security providers, such as Tivoli Access Manager for WebLogic, to seamlessly integrate their security functions into the BEA WebLogic Server architecture.

**Tivoli Access Manager Security Service Provider Interface components**

Tivoli Access Manager for WebLogic replaces the default security realm created with each BEA WebLogic Server secure domain and provides the following BEA WebLogic Server Security Providers:

- Authentication Provider
- Authorization Provider
- Role Mapping Provider

Tivoli Access Manager for WebLogic uses the default BEA WebLogic Server Credential Mapping security provider and the default keystore.

Each of the Providers listed above also contain a Management Bean (MBean) that enables configuration editing through the WebLogic console. The sections below detail the functionality supplied by each of these Providers and MBeans.

Tivoli Access Manager provides the following integration points with BEA WebLogic Server:

**Authentication Provider**

The Tivoli Access Manager for WebLogic Authentication Provider implements BEA WebLogic Server simple authentication. In simple authentication, a user attempts to authenticate to a BEA WebLogic Server using a username and password combination. This username and password are checked by Tivoli Access Manager using the Tivoli Access Manager Java runtime component.
Tivoli Access Manager for WebLogic also provides its own Login Module that is used to provide WebSEAL or Tivoli Access Manager Plug-in for Web Servers single sign-on functionality. Details of enabling single sign-on functionality is included in Chapter 4, “Enabling Single Signon,” on page 31.

The Authentication Provider for Tivoli Access Manager for WebLogic consists of several components:

- Authentication Provider
  Integrates the IBM Tivoli Access Manager for WebLogic Server Authentication Provider into the WebLogic Security Framework.

- Java Authentication and Authorization Service (JAAS) Login Module
  Performs simple and single sign-on authentication. JAAS Login Modules return a Subject populated with Principals as specified by the JAAS standard. Tivoli Access Manager for WebLogic provides its own Login Module, which uses the Tivoli Access Manager Java runtime component to authenticate to the Tivoli Access Manager authorization server.

- Authentication MBean
  Enables configuration of the Authentication Provider through the WebLogic console. It also allows users to perform user registry administration tasks, such as adding and deleting users using the Tivoli Access Manager for WebLogic console extension.

**Authorization Provider**

Authorization Providers supply an interface between BEA WebLogic Server and the external authorization service. The Authorization Provider determines whether access should be granted or denied to BEA WebLogic Server resources. The access decision is made using the PDP/Permission classes that are distributed with the Tivoli Access Manager Java runtime component.

The Authorization Provider for Tivoli Access Manager for WebLogic consists of the following components:

- Authorization Provider
  Integrates the Authorization Provider into the WebLogic Security Framework. In addition to controlling access to BEA WebLogic Server resources, the Tivoli Access Manager for WebLogic Authorization Provider handles the deployment of policy into the Tivoli Access Manager object space and the removal of policy from the Tivoli Access Manager object space.

- Authorization MBean
  Enables configuration of the Authorization Provider through the WebLogic console. It is also invoked for operations such as creating and deleting policy through the WebLogic console.

**Role Mapping Provider**

Role Mapping Providers are used to supply an interface between BEA WebLogic Server and the external authorization service that is being used to manage roles. The Role Mapping Provider focuses on roles rather than on policy which is the responsibility of the Authorization Provider.

The Role Mapping Provider consists of the following components:

- Role Mapping Provider
  Integrates the Role Mapping Provider into the WebLogic Security Framework. The Tivoli Access Manager for WebLogic Role Mapping Provider is responsible for the deployment and removal of roles.
• Role Mapping MBean.
  Enables the configuration of the Role Mapping Provider through the WebLogic console. It is also invoked for operations such as creating and updating role membership by deleting roles through the WebLogic console.

Policy and role deployment
Policy and roles can be defined in deployment descriptors or created through the WebLogic console. Upon deployment of J2EE applications, roles and policy defined within the application deployment descriptors are exported to the Tivoli Access Manager protected object space.

Although possible, it is not expected that policy creation will be performed using the Tivoli Access Manager administrative utility, pdadmin or the Tivoli Access Manager Web Portal Manager. Before starting a BEA WebLogic Server that is using Tivoli Access Manager for WebLogic, it is required that some default policy be created in Tivoli Access Manager. This is performed during Tivoli Access Manager for WebLogic configuration — details of Tivoli Access Manager for WebLogic configuration is covered in Chapter 3, “Configuration procedures,” on page 19.

Resources and roles
BEA WebLogic Server defines a number of different resource types, all of which are supported by Tivoli Access Manager for WebLogic. All resource types are considered the same within Tivoli Access Manager for WebLogic and so new resource types, created for future releases of BEA WebLogic Server, will be supported automatically.

The policies and roles defined for all resource types are stored in the Tivoli Access Manager protected object space in a uniform way.

The current list of supported BEA WebLogic Server resources that can be secured is:
• Administrative resources
• Application resources
• COM resources
• EIS resources
• EJB resources
• JDBC resources
• JMS resources
• Server resources
• URL resources
• Web Service resources

Resources are represented in the following format in the Tivoli Access Manager protected object space:
/WebAppServer/WLS/Resources/wls_domain/wls_realm/resource_type/Details

Roles are represented in the following format in the Tivoli Access Manager protected object space:
/WebAppServer/WLS/Roles/wls_domain/wls_realm/role_name/AppName

These Tivoli Access Manager protected object container names are fully configurable using the properties files configured with Tivoli Access Manager for
WebLogic. Thus all BEA WebLogic Server and other application servers can be configured into the same Tivoli Access Manager domain. This enables the creation of a centralized location for roles and policy for all application server types.

Using Tivoli Access Manager authentication

Tivoli Access Manager can be used to provide authentication for either external users or internal users. Authentication for external users relies on the single sign-on capabilities of either WebSEAL or the Tivoli Access Manager Plug-in for Web Servers. For optimum network security, each WebLogic Server that receives access requests from external users through either WebSEAL or the Tivoli Access Manager Plug-in for Web Servers should not accept access requests from internal users. The following sections describe how authentication is handled for both external and internal users.

Authenticating external users with WebSEAL

The diagram below displays the model for the processing of requests from external users for access to protected resources.

![Diagram](image)

Figure 1. Tivoli Access Manager provides single sign-on authentication for external users

The following list describes the processes as shown in figure above.

1. An external user requests access to a protected resource. The request is received by WebSEAL before entering the secure network of the enterprise.

2. WebSEAL intercepts the user request and authenticates the user in the Tivoli Access Manager secure domain.

   WebSEAL supports the following authentication methods: username and password, certificates, username and RSA SecureID, or a custom authentication mechanism.

   WebSEAL applies its own authorization decision based on the requested URL and the Tivoli Access Manager access policy. WebSEAL can apply considerations such as account validity, time-of-day, and authentication mechanism.

3. Once the user’s URL request has been authorized, WebSEAL forwards it to the WebLogic server. The request includes the external username and a special
password within the basic authentication header. The special password belongs to the \texttt{sso\_user}, and allows the Security Service Provider Interface to confirm WebSEAL as the origin of the request.

For more information about the \texttt{sso\_user}, see Chapter 3, “Configuration procedures,” on page 19.

4. The WebLogic server transparently passes the authenticated user identity and password to the Security Service Provider Interface.

5. The Security Service Provider Interface uses Tivoli Access Manager authentication services to verify that the password provided by WebSEAL is correct for the \texttt{sso\_user} described above. That is, this password provides the basis of trust that the request’s origin is WebSEAL.

The request is now ready for authorization.

**Authenticating internal users**

The diagram below displays the model for the processing of requests for access to protected resources by internal users who do not go through a WebSEAL or plug-in security:

![Diagram of Authenticating Internal Users](image)

\textit{Figure 2. Tivoli Access Manager Custom Realm provides authentication of internal users}

The following list describes the processes as shown in figure above.

1. An internal user requests access to a protected resource.
2. The WebLogic User Authentication module sends the user’s identity to the Security Service Provider Interface.
3. The Security Service Provider Interface sends the authentication request to the user registry.
   
   If authentication is successful, the Security Service Provider Interface returns the username to WebLogic Server as the authenticated user.
4. To authorize a request, BEA WebLogic Server queries the Tivoli Access Manager for WebLogic Authorization provider which determines whether the current authenticated user (perhaps unauth) is authorized to access the requested resource.
Access is determined by calls to the Tivoli Access Manager Authorization Server which selects the roles that are granted access to the resource and decides whether the current authenticated user is granted any of those roles.

Logging and auditing

Logging within Tivoli Access Manager for WebLogic is handled by the IBM JLog classes that are distributed with the Tivoli Access Manager Java runtime component. By using the Tivoli Access Manager for WebLogic and JLog properties files shipped with Tivoli Access Manager for WebLogic, the JLog classes can be configured to use the BEA WebLogic Server logging classes. This enables Tivoli Access Manager for WebLogic to log events directly to the WebLogic log files.

Reliability, Availability, and Scalability

Tivoli Access Manager for WebLogic uses the Tivoli Access Manager Java Runtime classes to manipulate the Tivoli Access Manager protected object database and the user registry. An internal Tivoli Access Manager for WebLogic cache provides performance enhancements for access decisions.

Tivoli Access Manager Java runtime classes support Tivoli Access Manager authorization server failover. If the primary authorization server crashes, failover to a secondary server occurs automatically.

The recommended environment setup is to use replicated aclds and the Tivoli Access Manager for WebLogic entitlements service.

Access decisions can be performed using either the Tivoli Access Manager Policy Server or the Tivoli Access Manager Authorization Server entitlement service that is shipped with Tivoli Access Manager for WebLogic.

Tivoli Access Manager Policy Server configuration should only be used in test environments due to a single point of failure and performance issues. The entitlement service is developed specifically for use in production environments. For further details refer to “Enabling the entitlements service on a Tivoli Access Manager Authorization server” on page 33.
Chapter 2. Installation instructions

This chapter contains the following topics:

- "Supported platforms"
- "Disk and memory requirements"
- "Software prerequisites" on page 10
- "Installing using the installation wizard" on page 11
- "Installing using native utilities" on page 14

Supported platforms

Tivoli Access Manager for WebLogic, Version 5.1, supports:

- BEA WebLogic Server version 7.0 SP2
- BEA WebLogic Server version 8.1 SP1

Tivoli Access Manager for WebLogic does not support the Custom Realm for this release. Instead, this integration supports the BEA WebLogic Server Security Service Provider Interface (SSPI).

Tivoli Access Manager for WebLogic is supported on the following operating systems:

- IBM AIX 5.1
- Sun Solaris 8 and 9
- Hewlett-Packard HP-UX 11.0 and 11i (BEA WebLogic Server version 7.0 only)
- Microsoft Windows 2000 Server and Advanced Server (Service Pack 3)

Note: Tivoli Access Manager for WebLogic supports systems running with the Java 2 Security Manager enabled. A Java policy file is provided with the software that includes the permissions necessary for specific codebases for Java 2 Security Manager to work.

Disk and memory requirements

Tivoli Access Manager for WebLogic has the following disk and memory requirements:

- 64 MB RAM, 128 MB recommended.
  This is the amount of memory needed in addition to the memory requirements specified by BEA WebLogic Server and by any other Tivoli Access Manager components. The additional 64 MB RAM is used to optimize caching performance.
  The amount of memory needed by other Tivoli Access Manager components will depend on which Tivoli Access Manager components are installed on the host system. For more information, see the IBM Tivoli Access Manager Base Installation Guide.
- 2 MB disk space, 4 MB recommended.
  This requirement is in addition to the disk space required by BEA WebLogic Server and by any other Tivoli Access Manager components.
- 5 MB disk space for log files.
This is in addition to the disk space required for the software components.

Software prerequisites

Successful installation of Tivoli Access Manager for WebLogic requires the prerequisites described in the following sections:

- “Tivoli Access Manager policy server”
- “Tivoli Access Manager WebSEAL or Tivoli Access Manager Plug-in for Web Servers”
- “BEA WebLogic Server” on page 11
- “Tivoli Access Manager Java runtime” on page 11

Tivoli Access Manager policy server

A Tivoli Access Manager secure domain must be established prior to installing Tivoli Access Manager for WebLogic.

The Tivoli Access Manager secure domain is established when you install the Tivoli Access Manager policy server. This policy server is distributed on the IBM Tivoli Access Manager Base CD for your operating system.

Typically, the Tivoli Access Manager policy server is installed on a system other than the system that hosts Tivoli Access Manager for WebLogic.

Tivoli Access Manager Authorization Server

The Tivoli Access Manager Authorization Server should be installed on the same host where BEA WebLogic Server and Tivoli Access Manager for WebLogic are installed.

The authorization server provides BEA WebLogic Server with access to the Tivoli Access Manager authorization service. The authorization server also acts as a logging and auditing collection server to store records of server activity.

Tivoli Access Manager WebSEAL or Tivoli Access Manager Plug-in for Web Servers

Tivoli Access Manager WebSEAL (WebSEAL) and Tivoli Access Manager Plug-in for Web Servers (the plug-in) provide Web-based security services that can be used by Tivoli Access Manager for WebLogic. When installed, these applications can be used to provide a BEA WebLogic Server single sign-on solution.

WebSEAL or the plug-in are not prerequisites for installing Tivoli Access Manager for WebLogic. They are necessary though if you require a single sign-on solution.

For installation instructions for WebSEAL or the plug-in, see the IBM Tivoli Access Manager for e-business Web Security Installation Guide.

When using WebSEAL or other proxy server to connect to BEA WebLogic Server, you should ensure that the proxy server is the single point of contact for users accessing BEA WebLogic Server protected resources. To restrict access, you will need to create a BEA WebLogic Server connection filter. A connection filter allows you to protect resources at the network level, rather than using roles to restrict access. Refer to the BEA WebLogic Server documentation for details on creating connection filters.
BEA WebLogic Server

BEA WebLogic Server must be installed and configured on the system that will host Tivoli Access Manager for WebLogic. BEA WebLogic Server is launched using the `startWebLogic` command.

BEA WebLogic Server is distributed with the necessary Java Runtime Environment on all supported platforms except AIX. Tivoli Access Manager for WebLogic uses this same Java Runtime Environment. Successful installation of BEA WebLogic Server satisfies the Tivoli Access Manager for WebLogic prerequisite for a Java Runtime Environment.

IBM Java Runtime Environment on AIX
On AIX systems, BEA WebLogic Server 7.0 requires IBM Java Runtime Environment Version 1.3 to be installed on the system that will host Tivoli Access Manager for WebLogic. On AIX systems, BEA WebLogic Server 8.1 requires IBM Java Runtime Environment Version 1.4 to be installed on the system that will host Tivoli Access Manager for WebLogic. Tivoli Access Manager for WebLogic uses these same versions of the Java Runtime Environment.

Tivoli Access Manager Java runtime
The Tivoli Access Manager Java runtime version 5.1 environment from the Tivoli Access Manager Base must be installed and configured on the system that will host Tivoli Access Manager for WebLogic.

The Tivoli Access Manager Java runtime environment provides Java-based authentication and authorization facilities. These Java classes extend the Java runtime environment that is used by BEA WebLogic Server.

The Tivoli Access Manager secure domain must be established prior to configuring the Tivoli Access Manager Java runtime environment on the system that will host Tivoli Access Manager for WebLogic.

The Tivoli Access Manager Java runtime environment is distributed with the IBM Tivoli Access Manager Base CD for each supported operating system. For installation instructions, see the `IBM Tivoli Access Manager Base Installation Guide`.

---

**Installing using the installation wizard**

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>This installation wizard is supported for the default installation location of BEA WebLogic Server, Version 7.0 only. If you are using BEA WebLogic Server, Version 8.1, follow the instructions in &quot;Installing using native utilities&quot; on page 14.</td>
</tr>
</tbody>
</table>

The `install_amwls` installation wizard simplifies the set up of a Tivoli Access Manager for WebLogic Server system by installing and configuring the following components in the appropriate order:

- Access Manager Java Runtime Environment
- Access Manager for WebLogic Server

To install and configure a Tivoli Access Manager for WebLogic Server system using the `install_amwls` wizard, follow these steps:
1. Ensure that you have already set up a Tivoli Access Manager registry server, a policy server, and an authorization server in your domain.
2. Ensure that all necessary operating system patches are installed. For information, see "Supported platforms" on page 9.
3. To view status and messages in a language other than English (default), you must install a language support package before running an installation wizard.
4. Ensure that BEA WebLogic Server is installed and configured on this machine and a WebLogic Server domain has been created.
5. On Windows systems, exit from all running programs.
6. Start the BEA WebLogic Server:
   
   **UNIX**
   
   `/WLS_install_dir/user_projects/domain_name/startWebLogic.sh`
   
   **Windows**
   
   `C:\WLS_install_dir\user_projects\domain_name\startWebLogic.cmd`
   
   7. Set the CLASSPATH and PATH variables and add WebLogic .jars to the CLASSPATH, and bin and lib directories by executing the following script from the BEA WebLogic Server `WebLogic_install_dir/server/bin` directory:
   
   **UNIX**
   
   `.setWLSEnv.sh`
   
   **Windows**
   
   `setWLSEnv.cmd`
   
   Ensure that the java executable shipped with BEA WebLogic Server is the first in the system path, prior to running the installation Wizard.
   
   8. Run the `install_amwls` program, located in the root directory on Tivoli Access Manager Web Security CD for AIX, HP-UX (BEA WebLogic Server 7.0 only), Solaris, and Windows platforms. If BEA WebLogic Server is not installed in the default location, you will need to use the following command to run the install Wizard:
   
   `install_amwls -is:javahome path`
   
   Where `path` is the location of the jre used to perform the installation using the Wizard.
   
   **Notes:**
   
   1. The `install_amwls.options.template` file can be used for silent installation or simply to override default installation values. Simply edit the file to include all of the values required.
   
   - To override default values use the command:
     
     `install_amwls -options install_amwls.options.template`
   
   - To perform a silent install use:
     
     `install_amwls -silent -options install_amwls.options.template`
   
   2. The installation wizard on non-english platforms may display incoherent text on the Welcome screen when using the JDKs shipped with BEA WebLogic Server. This display issue does not affect the actual software installation. If you wish to fix this problem, install the IBM JDK 1.3.1 and use it to run `install_amwls`.
   
   The installation wizard begins by prompting you for configuration information as described in "install_amwls options" on page 13. On Windows systems only, ensure that you accept the default installation directory for Tivoli Access Manager for WebLogic.
**Note:** After you supply this information (or accept default values), the components are installed and configured without further intervention.

A summary screen is displayed at the end of the install Wizard that shows the components installed and what configurations were attempted and whether or not they succeeded. If the installation succeeds but the configuration fails, you can try to configure Tivoli Access Manager for WebLogic manually by following the steps in Chapter 3, “Configuration procedures,” on page 19, otherwise continue with the following steps.

9. Stop the BEA WebLogic Server.
10. Check that the installation copied the file, AMSSPIProviders.jar, into the /bea_install_dir/weblogic/server/lib/mbeatypes directory. If the file does not exist on this directory, copy it manually from /amwls_install_dir/lib.
11. Set the CLASSPATH for the startWebLogic command by following the instructions in “Part 2: Setting CLASSPATH for startWebLogic” on page 20. Create and configure the Tivoli Access Manager realm. For instructions, see “Part 4: Configuring the Tivoli Access Manager Realm” on page 24.
12. Restart the BEA WebLogic Server using the WebLogic console.
13. If you want to use Tivoli Access Manager WebSEAL to provide single sign-on services for BEA WebLogic Server, follow the instructions at “Part 5: Configuring for BEA WebLogic Server single signon” on page 26.
14. Test the installation and configuration to ensure that the Tivoli Access Manager for WebLogic has been correctly configured against Tivoli Access Manager registry by completing the steps at “Part 7: Testing the configuration” on page 29.

**install_amwls options**

The following are the options displayed when running `install_amwls`.

<table>
<thead>
<tr>
<th>Configuration Options</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote ACL User *</td>
<td>The Tivoli Access Manager principal that is created for communication with the authorization server.</td>
<td></td>
</tr>
<tr>
<td>sec_master password *</td>
<td>The Tivoli Access Manager administrator password.</td>
<td></td>
</tr>
<tr>
<td>Policy server host name *</td>
<td>The fully qualified host name of the policy server. For example: pdmgr.tivoli.com</td>
<td></td>
</tr>
<tr>
<td>Policy server port number *</td>
<td>The port number on which the policy server listens for requests. The default port number is 7135.</td>
<td>7135</td>
</tr>
<tr>
<td>Authorization server host name*</td>
<td>The Tivoli Access Manager authorization server host name.</td>
<td></td>
</tr>
<tr>
<td>Authorization server port number*</td>
<td>The authorization server port number.</td>
<td>7136</td>
</tr>
<tr>
<td>Deploys the AMWLS5.1 console extension when set to true</td>
<td>Administrator of the BEA WebLogic Server domain. This user would have been established when you created your WebLogic domain.</td>
<td>true</td>
</tr>
<tr>
<td>The WebLogic Domain Administrator *</td>
<td>Administrator of the BEA WebLogic Server domain. This user would have been established when you created your WebLogic domain.</td>
<td></td>
</tr>
</tbody>
</table>
Installing using native utilities

Complete the instructions in the section for your operating system:

- “Installing on AIX”
- “Installing on HP-UX” on page 15
- “Installing on Solaris” on page 15
- “Installing on Windows” on page 16

Note: Before installing Tivoli Access Manager for WebLogic be sure to stop the BEA WebLogic Server and then restart it after installation is complete.

Installing on AIX

The Tivoli Access Manager for WebLogic installation separates file extraction from package configuration. Use installp to install software packages on AIX. Then configure Tivoli Access Manager for WebLogic manually.

Note: If you have already installed and configured Tivoli Access Manager for WebLogic and need to reinstall it, you must first unconfigure and remove the Tivoli Access Manager for WebLogic package. See “Removing from AIX” on page 42.

To install Tivoli Access Manager for WebLogic on AIX complete the following instructions:

1. Log in as root.
2. Verify that the software prerequisites, including the required components from the Tivoli Access Manager base, have been satisfied. See “Software prerequisites” on page 10.
3. Insert the IBM Tivoli Access Manager Web Security for AIX CD into the CD drive.
4. Enter the following command at a shell prompt:
   ```
   installp -acgNXd cd_mount_point/usr/sys/inst.images PDWLS
   ```
   Note: Check that the installation copied the file, AMSSPIProviders.jar, into the /bea_install_dir/weblogic/server/lib/mbeantypes directory. If the file does not exist on this directory copy it manually from /amwls_install_dir/lib.
Installing on HP-UX

Attention
When installed on the HP-UX platform, Tivoli Access Manager for WebLogic is only supported for BEA WebLogic Server version 7.0.

If you have already installed and configured Tivoli Access Manager for WebLogic and need to reinstall it, you must first unconfigure and remove it. See “Removing from HP-UX” on page 42.

To install Tivoli Access Manager for WebLogic on HP-UX, complete the following steps:

1. Log in as user root.
2. Verify that the software prerequisites, including the required components from the Tivoli Access Manager base, have been satisfied. See “Software prerequisites” on page 10.
3. Start pfs_mountd and then pfsd in the background, if they are not running.
   Mount the CD with the pfs_mount command. For example, enter the following:
   ```
   /usr/sbin/pfs_mount /dev/dsk/c0t0d0 /cd-rom
   ```
   where /dev/dsk/c0t0d0 is the CD device and /cd-rom is the mount point.
4. Enter the following command to install the Tivoli Access Manager for WebLogic package:
   ```
   # swinstall -s /cd_rom/hp PDWLS
   ```
   A message is displayed indicating that the analysis phase has succeeded.
   Another message is displayed indicating that the execution phase is beginning.
   Files are extracted from the CD and installed on the hard disk. A message is displayed indicating that the execution phase has succeeded. The swinstall utility exits.

   **Note:** Check that the installation copied the file, AMSSPIProviders.jar, into the /bea_install_dir/weblogic/server/lib/mbeantypes directory. If the file does not exist on this directory copy it manually from /amwls_install_dir/lib.


Installing on Solaris

The Tivoli Access Manager for WebLogic installation separates file extraction from package configuration. Use pkgadd to install software packages on Solaris Operating Environment (hereinafter referred to as Solaris). Then configure Tivoli Access Manager for WebLogic manually.

**Note:** If you have already installed and configured Tivoli Access Manager for WebLogic and need to reinstall it, you must first unconfigure and remove it. See “Removing from Solaris” on page 41.

To install Tivoli Access Manager for WebLogic on Solaris, complete the following instructions:
1. Log in as user root.

2. Verify that the software prerequisites, including the required components from the Tivoli Access Manager base, have been satisfied. See “Software prerequisites” on page 10.

3. Insert the IBM Tivoli Access Manager Web Security for Solaris CD.

4. Run the following command to install the software:

   `pkgadd -d /cdrom/cdrom0/solaris -a /cdrom/solaris/pddefault PDWLS`

   where:

   `-d /cdrom/cdrom0/solaris` Specifies the location of the package.

   `-a /cdrom/cdrom0/solaris/pddefault` Specifies the location of the installation administration script.

   When the installation process is complete for each package, the following message is displayed:

   Installation of package successful.

   **Note:** Check that the installation copied the file, `AMSSP1Providers.jar`, into the `/bea_install_dir/weblogic/server/lib/mbeantypes` directory. If the file does not exist on this directory copy it manually from `/amwls_install_dir/lib`.


**Installing on Windows**

The Tivoli Access Manager for WebLogic installation separates file extraction from package configuration. Use an InstallShield `setup.exe` to install the Tivoli Access Manager for WebLogic files. When InstallShield completes, configure Tivoli Access Manager for WebLogic using the instructions in Chapter 3, “Configuration procedures,” on page 19.

**Note:** If you have already installed and configured Tivoli Access Manager for WebLogic and need to reinstall it, you must first unconfigure and remove it. See “Removing from Windows” on page 41.

To install Tivoli Access Manager for WebLogic on Windows complete the following instructions:

1. Log in to the Windows domain as a user with Windows administrator privileges.

2. Verify that the software prerequisites, including the required components from the Tivoli Access Manager base, have been satisfied. See “Software prerequisites” on page 10.

3. Insert the IBM Tivoli Access Manager Web Security for Windows CD into the CD drive.

4. Run the Tivoli Access Manager for WebLogic InstallShield setup program by double-clicking on the following file, where the letter E: in the following command represents the CD drive:

   `E:\Windows\PolicyDirector\Disk Images\Disk1\PDWLS\Disk Images\Disk1\setup.exe`

   The Choose SetupDirector Disk Images\Disk1\PDWLS\Disk Images\Disk1\setup.exe

   The Choose SetupDirector Disk Images\Disk1\PDWLS\Disk Images\Disk1\setup.exe

   The Choose SetupDirector Disk Images\Disk1\PDWLS\Disk Images\Disk1\setup.exe

5. Select the appropriate language and click OK.
The InstallShield program starts and the Welcome window opens.

6. Click **Next**.
The License Agreement window opens.

7. Read the license agreement and click **Yes** if you accept the terms and conditions.
The Choose Destination Location window opens.

8. Accept the default location or browse for an alternative location. Click **Next**.
The Start Copying Files window opens.

9. Ensure the displayed installation location is correct and click **Next**.
The files are extracted to the disk. A message displays indicating that the files have been installed.

10. Click **Finish** to exit the setup program.

11. Check that the installation copied the file, AMSSPIProviders.jar, into the
    `c:\bea_install_dir\weblogic\server\lib\mbeantypes` directory. If the file does not exist on this directory copy it manually from `c:\amwls_install_dir\lib`.

12. Next, configure Tivoli Access Manager for WebLogic. Go to [Chapter 3](#). **Configuration procedures,” on page 19**.
Chapter 3. Configuration procedures

To configure Tivoli Access Manager for WebLogic, complete the instructions in each of the following sections:

- “Part 1: Configuring the Tivoli Access Manager Java runtime environment”
- “Part 2: Setting CLASSPATH for startWebLogic” on page 20
- “Part 3: Configuring Tivoli Access Manager for WebLogic” on page 21
- “Part 4: Configuring the Tivoli Access Manager Realm” on page 24
- “Part 5: Configuring for BEA WebLogic Server single signon” on page 26
- “Part 6: Configuring Tivoli Access Manager for WebLogic in BEA WebLogic Server multi-server environments including clustered environments” on page 28
- “Part 7: Testing the configuration” on page 29

Note: The instructions in this chapter assume you have installed Tivoli Access Manager for WebLogic and the prerequisite software, including configuration of the Tivoli Access Manager base components. If you have not installed the software, install it now by following the instructions in Chapter 2, “Installation instructions,” on page 9.

Part 1: Configuring the Tivoli Access Manager Java runtime environment

The Tivoli Access Manager Java runtime environment is a prerequisite for Tivoli Access Manager for WebLogic. The Java runtime component must be configured correctly before a BEA WebLogic Server realm can be configured. Use the Tivoli Access Manager utility pdjrtecfg to update the Java Runtime Environment used by the BEA WebLogic Server. Also, if the system contains multiple Java runtimes, ensure that the Java Runtime Environment used by BEA WebLogic Server is used to execute the pdjrtecfg utility.

1. Verify that the Tivoli Access Manager Base Java runtime environment has been installed.
   
   For more information see “Software prerequisites” on page 10.

2. Set the CLASSPATH and PATH variables and add WebLogic jars to the CLASSPATH, and bin and 11b directories by executing the following script from the BEA WebLogic Server WebLogic_install_dir/server/bin directory:

   UNIX  .setWLSEnv.sh
   
   Windows
   
   setWLSEnv.cmd

   Ensure that the java executable shipped with BEA WebLogic Server is first in the system path prior to running the ezInstall.

3. Tivoli Access Manager Java Runtime Environment needs to be configured against the JDK that is shipped and installed with BEA WebLogic Server. To do this:

   a. Change directory to the sbin directory in the Tivoli Access Manager installation path. For example:

   UNIX: /opt/PolicyDirector/sbin
   Windows: C:\Program Files\Tivoli\Policy Director\sbin
b. Execute the `pdrjtecfg` command as follows:

```
pdrjtecfg -action config -host policy_server_name -java_home java_location
```

Where `java_location` is the directory location of the BEA WebLogic Server Java Runtime Environment. This will be:

**Windows**

BEA WebLogic Server version 7.0

```
c:\\bea\\jdk131_ob\\jre
```

BEA WebLogic Server version 8.1

```
c:\\bea\\jdk141\\jre
```

**Solaris, HP-UX**

```
/usr/local/bea/jdk141_03
```

**AIX**

On AIX systems, BEA WebLogic Server 7.0 requires IBM Java Runtime Environment Version 1.3 and BEA WebLogic Server 8.1 requires IBM Java Runtime Environment Version 1.4. The `-java_home` option in the `pdrjtecfg` command should be set to the installation location of the JRE on your AIX machine. BEA WebLogic Server version 7.0

```
/usr/java131
```

BEA WebLogic Server version 8.1

```
/usr/java14
```

**Notes:**

1. The `pdrjtecfg` utility on BEA WebLogic Server 8.1 installations replaces `jsse.jar` on the `jre/lib` directory. This file is reinstated when the Tivoli Access Manager Java Runtime is unconfigured.

2. When configuring Sun v1.4 JRE do not run `pdrjtecfg` in interactive mode or use the `pdconfig` utility to configure the JRE as the configuration will fail.

For more information on the use of `pdrjtecfg`, see the reference page for it in the IBM Tivoli Access Manager Base Installation Guide.

---

**Part 2: Setting CLASSPATH for startWebLogic**

**Note:** Before executing these configuration steps, ensure you have created a WebLogic domain.

The `startWebLogic` command is used to start the WebLogic Server. You need to modify the CLASSPATH environment variable to enable `startWebLogic` to access and load the correct Java classes.

Complete the following instructions:

1. If the WebLogic Server is running, stop it now.
2. Add the following file names to the CLASSPATH variable of the `startWebLogic` command:

UNIX
The `startWebLogic` command is located in the directory of the installed domain of the BEA WebLogic Server. In a standard installation this is:

**UNIX**  `/WebLogic_install_directory/user_projects/domain_name`

**Windows**  `C:\WebLogic_install_directory\user_projects\domain_name`

The variable `domain_name` is the name you selected when you created your BEA WebLogic Server domain.

3. If you are using the default language (English) skip this step.

If you are using a language pack to support a language other than the default (English), you must add the following path to the CLASSPATH defined in the `startWebLogic` script:

**UNIX**  `/opt/pdwls/nls/java/com/tivoli/amwls/sspi/nls`

**Windows**  `C:\Progra~1\Tivoli\pdwls\nls\java\com\tivoli\amwls\sspi\nls`

**Note:** The addition of this directory enables access to the resource bundles that are installed in `/opt/pdwls/nls/java/com/tivoli/amwls/sspi/nls/` by the language pack installation.

---

**Part 3: Configuring Tivoli Access Manager for WebLogic**

Configuration of Tivoli Access Manager for WebLogic can be done either from the command line or using the Tivoli Access Manager Console Extension Web Application. Details of both of these options are included in the sections below.

A BEA WebLogic Server domain should be created prior to executing these instructions.

Data entered when configuring Tivoli Access Manager for WebLogic and creating the realm is stored in properties files. These properties files can be used to change the behavior of Tivoli Access Manager for WebLogic. For more information refer to Appendix A, “Properties Files Reference,” on page 45.

---

**Configuring Tivoli Access Manager for WebLogic using the Console Extension Web application**

1. Start the BEA WebLogic Server:

   **UNIX**  `/WLS_install_dir/user_projects/domain_name/startWebLogic.sh`

   **Windows**  `C:\WLS_install_dir\user_projects\domain_name\startWebLogic.cmd`

2. Open a Web browser and connect to the BEA WebLogic console on the machine hosting BEA WebLogic. That is:

   `http://WebLogic_server_name:7001/console`
7001 is the default BEA WebLogic Server port number. This value is configurable.

3. The BEA WebLogic Server logon screen is displayed. Logon as a BEA WebLogic Server user with administrator privileges.

4. Before configuring Tivoli Access Manager for WebLogic server and creating the Tivoli Access Manager realm you will need to deploy the Tivoli Access Manager Console Extension Web Application that provides a Web interface to the configuration tasks. To deploy this Web application:
   
a. From the BEA WebLogic Server home page, within the Domain Configurations banner, select Web Applications.
   
b. Select the Configure a New Web Application link.
   
c. Select the Upload it through your browser link.
   
d. Browse for the application amwls_install_dir\lib\AMWLSCsoleExtension.war. Click Upload.
   
e. Click on the Select link for AMWLSCsoleExtension.war.
   
f. Select the deployment target and click Configure and Display.
   
   To check that the Console Extension Web Application has been successfully deployed, expand the Deployments folder in the left-hand screen pane. Expand the Web Applications folder and AMWLSCsoleExtensions should be displayed within the list. Also deploying the Console Web Application Extension will have added a Tivoli Access Manager icon in the BEA WebLogic Server navigation pane displayed on the left of the console window.

5. To configure the Tivoli Access Manager domain click on the Access Manager icon in the BEA WebLogic Server navigation pane.

6. The configuration screen is displayed. Enter all required information and any optional parameters. Refer to the tables below for guidance on the information to enter.

   The options available to the config action are listed in the tables below. The first table contains required options. The second table contains optional options.

### Required Option Name | Description
---|---
domain_admin | WebLogic Domain Administrator
domain_admin_pwd | WebLogic Domain Administrator password
remote_acl_user | Tivoli Access Manager principal that is created for the authorization server
sec_master_pass | Tivoli Access Manager sec_master administrator password
pdmgrd_host | Tivoli Access Manager policy server host name.
pdacld_host | Tivoli Access Manager authorization server host name.

**Note:** Passwords do not have to be entered and will instead be prompted for prior to the action being performed. This is to prevent passwords from remaining in command history.

The following table contains optional options to the config action.

### Option Name | Description
---|---
wls_server_url | Specifies the URL for the local WebLogic Server. The default is t3://localhost:7001.
pdmgrd_port | Tivoli Access Manager policy server port number.
pdacld_port | Tivoli Access Manager authorization server port number.
---|---
am_domain | Specifies the name of the Tivoli Access Manager domain. The default is Default.
amwls_home | Specifies the path to the Tivoli Access Manager for WebLogic Server installation directory.

Click **Apply**.

7. If the configuration was successful, a list of the Tivoli Access Manager for WebLogic Server parameters is displayed in the right-hand pane.

The Tivoli Access Manager realm can now be configured. See "Part 4: Configuring the Tivoli Access Manager Realm" on page 24.

**Configuring Tivoli Access Manager for WebLogic from the command line**

1. Start the BEA WebLogic Server:

   UNIX

   `/WLS_install_dir/user_projects/domain_name/startWebLogic.sh`

   Windows

   `C:\WLS_install_dir\user_projects\domain_name\startWebLogic.cmd`

2. Execute the following command to configure Tivoli Access Manager for WebLogic.

   **Note:** If Tivoli Access Manager for WebLogic was not installed into the recommended location during the file extraction (as described in the previous chapter), be sure to set the `AMSSPI_DIR` variable in the `AMWLSConfigure` script to the location of the actual installation directory. Similarly, if WebLogic is not installed in the default location update the `WLS_JAR` variable with the correct location of `WebLogic.jar` in the `ALWLSConfigure` script.

   UNIX

   `install-dir/sbin/AMWLSConfigure.sh`

   Windows

   `install-dir\sbin\AMWLSConfigure.bat`

The command line syntax for the **AMWLSConfigure** Java application to configure Tivoli Access Manager for WebLogic is:

- **AMWLSConfigure -action config [options ...]**
  
  Configures Tivoli Access Manager for WebLogic.

- **AMWLSConfigure -help [action]**
  
  Displays the required and optional values to pass into **AMSSPIConfigure**.

The options available to the config action are listed in the tables below. The first table contains **required** options. The second table contains **optional** options.

<table>
<thead>
<tr>
<th>Required Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain_admin</td>
<td>WebLogic Domain Administrator</td>
</tr>
<tr>
<td>domain_admin_pwd</td>
<td>WebLogic Domain Administrator password</td>
</tr>
<tr>
<td>remote_acl_user</td>
<td>Tivoli Access Manager principal that is created for the authorization server</td>
</tr>
</tbody>
</table>
sec_master_pass | Tivoli Access Manager sec_master administrator password
pdmgrd_host | Tivoli Access Manager policy server host name.
pdacld_host | Tivoli Access Manager authorization server host name.

**Note:** Passwords do not have to be entered and will instead be prompted for prior to the action being performed. This is to prevent passwords from remaining in command history.

The following table contains optional options to the config action.

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deploy_extension</td>
<td>Deploys the Tivoli Access Manager for Web Logic Server console extension when set to true. The default value is true.</td>
</tr>
<tr>
<td>wls_server_url</td>
<td>Specifies the URL for the local WebLogic Server. The default is t3://localhost:7001.</td>
</tr>
<tr>
<td>pdmgrd_port</td>
<td>Tivoli Access Manager policy server port number.</td>
</tr>
<tr>
<td>pdacld_port</td>
<td>Tivoli Access Manager authorization server port number.</td>
</tr>
<tr>
<td>am_domain</td>
<td>Specifies the name of the Tivoli Access Manager domain. The default is Default.</td>
</tr>
<tr>
<td>amwls_home</td>
<td>Specifies the path to the Tivoli Access Manager for WebLogic Server installation directory.</td>
</tr>
<tr>
<td>verbose</td>
<td>A boolean value that enables or disables verbose output. The default value is false.</td>
</tr>
</tbody>
</table>

The Tivoli Access Manager realm now needs to be configured.

**Part 4: Configuring the Tivoli Access Manager Realm**

**Configuring the Tivoli Access Manager Realm using the Console Extension Web application**

Once Tivoli Access Manager for WebLogic Server has been configured to provide security for BEA WebLogic Server, you need to create a realm to associate with Tivoli Access Manager security. To do this:

1. Expand the Access Manager icon in the left-hand screen pane and click the Realm icon.
2. The Create Realm screen is displayed. Enter all required variables. Click Apply.
3. To configure BEA WebLogic Server 7.0 to use the Tivoli Access Manager realm created above:
   a. Select the icon related to your domain in the BEA WebLogic Server navigation pane.
   b. The Domain Configuration screen is displayed. Select the Security tab.
   c. From the General tab, use the Default Realm drop-down list, to select the realm created in the steps above. Click Apply.

To configure BEA WebLogic Server 8.1 to use the Tivoli Access Manager realm created above, use the Security tab on the BEA WebLogic Server console to set the default realm.

4. Restart the BEA WebLogic Server.
5. To test that the new Access manager realm is functioning correctly, the Users and Groups icons, within the Access Manager folder in the right-hand screen pane should contain entries from the Tivoli Access Manager user registry.

Note: If you specified an SSO user that already exists but entered the incorrect password for the existing user, the create realm action will succeed but SSO will be disabled. In this situation, SSO can be easily enabled by updating the appropriate entries in the Tivoli Access Manager for WebLogic rbpf.properties file. See Appendix A, “Properties Files Reference,” on page 45 for details on rbpf.properties.

Configuring the Tivoli Access Manager Realm from the command line

1. Execute the following command to create the Tivoli Access Manager for WebLogic realm.

Note: If Tivoli Access Manager for WebLogic was not installed into the recommended location during the file extraction (as described in the previous chapter), be sure to set the AMSSPI_DIR variable in the AMWLSConfigure script to the location of the actual installation directory. Similarly, if WebLogic is not installed in the default location or you are using WebLogic version 8.1, update the WLS_JAR variable with the correct location of WebLogic.jar in the ALWLSConfigure script.

UNIX  
install-dir/sbin/AMWLSConfigure.sh

Windows
install-dir\sbin\AMWLSConfigure.bat

The command line syntax for the AMWLSConfigure Java application to configure Tivoli Access Manager for WebLogic is:

• AMWLSConfigure -action create_realm [options ...]
  Creates the Tivoli Access Manager for WebLogic realm.
• AMWLSConfigure -help [action]
  Displays the required and optional values to pass into AMSSPIConfigure.

The options available to the create_realm action are listed in the tables below. The first table contains required options. The second table contains optional options.

<table>
<thead>
<tr>
<th>Required Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>realm_name</td>
<td>Specifies the name of the WLS realm being created.</td>
</tr>
<tr>
<td>domain_admin_pwd</td>
<td>Specifies the WebLogic domain administrator password.</td>
</tr>
<tr>
<td>user_dn_suffix</td>
<td>Specifies the distinguished name (DN) suffix to use when creating users through the console extension Web application.</td>
</tr>
<tr>
<td>group_dn_suffix</td>
<td>Specifies the distinguished name (DN) suffix to use when creating groups through the console extension Web application.</td>
</tr>
<tr>
<td>admin_group</td>
<td>Specifies the Tivoli Access Manager group to use for internal configuration purposes.</td>
</tr>
</tbody>
</table>
Note: Passwords do not have to be entered and will instead be prompted for prior to the action being performed. This is to prevent passwords from remaining in command history.

The following table contains optional options to the create_realm action.

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_dn_prefix</td>
<td>Specifies the distinguished name (DN) prefix to use when creating users through the console extension Web application.</td>
</tr>
<tr>
<td>group_dn_prefix</td>
<td>Specifies the distinguished name (DN) prefix to use when creating groups through the console extension Web application.</td>
</tr>
<tr>
<td>sso_enabled</td>
<td>Enables single sign-on support when set to true. The default value is false.</td>
</tr>
<tr>
<td>sso_user</td>
<td>Specifies the user for creating the single sign-on trust association with Tivoli Access Manager.</td>
</tr>
<tr>
<td>sso_pwd</td>
<td>Specifies the password for the single sign-on user.</td>
</tr>
<tr>
<td>verbose</td>
<td>A boolean value that enables or disables verbose output. The default value is false</td>
</tr>
</tbody>
</table>

2. To configure BEA WebLogic Server 7.0 to use the Tivoli Access Manager realm created above:

   a. Open a Web browser and connect to the BEA WebLogic console on the machine hosting BEA WebLogic. That is:
      
      \[
      \text{http://WebLogic_server_name:7001/console}
      \]

      7001 is the default BEA WebLogic Server port number, the value is configurable.

   b. The BEA WebLogic Server logon screen is displayed. Logon as a user with administrator privileges.

   c. Select the icon related to your domain in the BEA WebLogic Server navigation pane.

   d. The Domain Configuration screen is displayed. Select the Security tab.

   e. From the General tab, use the Default Realm drop-down list, to select the realm created in the steps above. Click Apply.

To configure BEA WebLogic Server 8.1 to use the Tivoli Access Manager realm created above, use the Security tab on the BEA WebLogic Server console to set the default domain.

3. Restart the BEA WebLogic Server.

4. To test that the new Access manager realm is functioning correctly, the Users and Groups icons, within the Access Manager folder in the left-hand screen pane should contain entries from the Tivoli Access manager user registry.

---

**Part 5: Configuring for BEA WebLogic Server single signon**

This section guides you through the process of configuring single signon to BEA WebLogic Server using either WebSEAL or Tivoli Access Manager Plug-in for Web Servers. If you do not want to implement single signon functionality you can ignore this section.

WebSEAL and Tivoli Access Manager Plug-in for Web Servers implement security and single signon in different ways and employ different system architectures. For information on installing WebSEAL and the plug-in for Web servers refer to the
Configuring single signon using a WebSEAL junction

To provide a single signon capability for BEA WebLogic Server using WebSEAL, complete the following steps on the system that hosts the WebSEAL server:

1. Open the WebSEAL configuration file, webseald.conf.
2. Set the following configuration item:
   
   ```
   basicauth-dummy-pwd = $ss_pwd
   ```

   This password must match that of the $ss_pwd field enabled during the create realm action.
3. Stop and restart WebSEAL, to make the configuration change take effect.
4. Use the `pdadmin` command to create a WebSEAL junction.

   **Note:** This step can be performed on any machine in the Tivoli Access Manager secure domain. You do not have to execute it on a WebSEAL system. For example, you could run it on the Tivoli Access Manager policy server system.

   Be sure to use the `-b` option to supply the junction target URL. This is required for single sign-on.

   For example, enter the following command as one continuous command line:

   ```
   pdadmin> server task webseald_server_name create -t tcp -p WebLogic_Server_listen_port -h WebLogic_Server -b supply junction_target
   ```

   The following table defines the variables in the above `pdadmin` command:

   **Table 2. Options for the pdadmin command**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webseald_server_name</td>
<td>Name of the WebSEAL server. The name consists of two parts: <code>webseald-WebSEAL_server_instance</code>. Use your system's host name for <code>WebSEAL_server_instance</code>. For example, if the host machine name is cruz, the <code>webseald_server_name</code> would be: webseald-cruz Note: If you have installed multiple instances of WebSEAL on the same server, you need to specify the server instance also. For instructions on creating junctions with multiple server instances, see the IBM Tivoli Access Manager for e-business WebSEAL Administration Guide.</td>
</tr>
<tr>
<td>WebLogic_Server</td>
<td>The host name of the BEA WebLogic Server.</td>
</tr>
</tbody>
</table>
Table 2. Options for the pdadmin command (continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebLogic_Server_listen_port</td>
<td>The port on which the BEA WebLogic Server is listening. The default is 7001.</td>
</tr>
<tr>
<td>-b supply</td>
<td>Required for single sign-on. Ensures WebSEAL passes the dummy password.</td>
</tr>
<tr>
<td>junction_target</td>
<td>The URL target of the junction</td>
</tr>
</tbody>
</table>

For complete information on creating and using WebSEAL junctions, see the IBM Tivoli Access Manager for e-business WebSEAL Administration Guide.

Configuring single signon using Tivoli Access Manager Plug-in for Web Servers

For single signon to work correctly, Tivoli Access Manager Plug-in for Web Servers needs to be configured to pass the correct information in the Basic Authentication header to IBM Tivoli Access Manager for WebLogic Server. For this to occur, Basic Authentication needs to be configured as a post-authorization module in the plug-in configuration file.

Edit the pdwebpi.conf configuration file located on the plug-in install_dir/etc directory, and add the following value to the [common-modules] stanza:

```
[common-modules]
post-authzn = BA
```

You then set the add-hdr and supply-password parameters in the [BA] stanza to BA and the password of the sso_user respectively. That is:

```
[BA]
add-hdr = supply
supply-password = sso_pwd
```

For more information on configuring Tivoli Access Manager Plug-in for Web Servers refer to the IBM Tivoli Plug-in for Web Servers Integration Guide.

Part 6: Configuring Tivoli Access Manager for WebLogic in BEA WebLogic Server multi-server environments including clustered environments

This section is for those architectures where BEA WebLogic Server is set up in a multi-server environment or clustered environment. To configure Tivoli Access Manager for WebLogic in BEA WebLogic Server multi-server environments including clustered environments:


2. Enable Tivoli Access Manager for WebLogic on managed servers, including cluster members by copying the Tivoli Access Manager for WebLogic properties from the administration server for the domain to each of the destination machines (managed servers). The properties files are located in, BEA_WLS_HOME/jdk_location/jre/amwls/ and should be copied to the same location on each of the managed servers.
Part 7: Testing the configuration

Verify that the Tivoli Access Manager for WebLogic has been correctly configured against Tivoli Access Manager registry by completing the following steps:

1. Use the BEA WebLogic Server console to create and validate a new test user.
2. Execute the following `pdadmin` command:
   ```
   pdadmin> user show test_user
   ```
   - Verify that account-valid is yes.
   - Verify that password-valid is yes.

The Tivoli Access Manager for WebLogic single sign-on solution allows a single authentication step through WebSEAL that transparently authenticates the user to the BEA WebLogic Server. You can confirm that authentication is configured correctly by running the demonstration application. The demonstration application is described in “Using the demonstration application” on page 35.
Chapter 4. Enabling Single Signon

Single sign-on with Tivoli Access Manager WebSEAL

Tivoli Access Manager for WebLogic supports Web single signon from other Tivoli Access Manager products such as Tivoli Access Manager WebSEAL, Tivoli Access Manager Plug-in for Web Servers and the Tivoli Access Manager Plug-in for Edge Server.

The trust relationship between WebSEAL and BEA WebLogic Server is achieved using a configured HTTP Basic Authentication dummy password. This same method was employed to perform single sign-on in previous Tivoli Access Manager for BEA WebLogic Server products implementing the Custom Security Realm interface.

The Tivoli Access Manager HTTP reverse proxy (for example, WebSEAL) is configured to pass the user name and a known single sign-on secret password. This secret password is used to determine if the reverse proxy is trusted. After the Tivoli Access Manager Authorization Server verifies the password, credentials are acquired for the user that is requesting the resource.

The figure below shows the details of how the trust relationship is established.

![Diagram of Single Sign-on using Tivoli Access Manager WebSEAL](image)

*Figure 3. Single sign-on using Tivoli Access Manager WebSEAL*

The figure above shows the following steps:

1. User authenticates to WebSEAL using any authentication mechanism supported by WebSEAL (for example, username/password or client certificate). User then submits a request for a BEA WebLogic Server resource.

2. WebSEAL has been configured with a junction to the BEA WebLogic Server using the -b supp1y option. WebSEAL passes the request to BEA WebLogic Server with a Basic Authentication header that contains:
The WebSEAL authenticated user-ID (**user-1** in the diagram)
- The value of basicauth-dummy-passwd in webseald.conf. This is the secret password that was referred to above.

3. BEA WebLogic Server passes the user-ID and secret password to the Tivoli Access Manager for WebLogic Authentication Provider for verification.

4. The Tivoli Access Manager for WebLogic Login Module uses Tivoli Access Manager to verify that the given password is for the Tivoli Access Manager for WebLogic configured WebSEAL single sign-on user. Verification of this password provides the trust relationship between WebSEAL and BEA WebLogic Server.

If step 4 is successful, the Tivoli Access Manager for WebLogic Authentication Provider authenticates the given user-ID to BEA WebLogic Server. Note that the authentication of the configured WebSEAL single sign-on user using the secret password (**ws-passwd** in diagram) is only performed once because it is cached in the Tivoli Access Manager for WebLogic Login Module. This cache is configurable and can be turned off.

SSO can be setup during realm creation, however, to manually enable SSO Tivoli Access Manager for WebLogic:

1. Create the SSO user.

2. In the amsspi.properties Tivoli Access Manager for WebLogic configuration file set:

```java
com.tivoli.amwls.sspi.Authentication.ssoEnabled = true
com.tivoli.amwls.sspi.Authentication.ssoTrustId = sso_username
```
Chapter 5. Administration tasks

This chapter contains the following information about Tivoli Access Manager for WebLogic:

- “Enabling the entitlements service on a Tivoli Access Manager Authorization server”
- “User and group management with Tivoli Access Manager for WebLogic” on page 34
- “Using the demonstration application” on page 35
- “Usage tips” on page 36
- “Three strikes logon policy” on page 37
- “Deleting the Tivoli Access Manager Realm” on page 38
- “Unconfiguring Tivoli Access Manager for WebLogic” on page 39
- “Troubleshooting tips” on page 39
- “Limitations” on page 40

Enabling the entitlements service on a Tivoli Access Manager Authorization server

By default, Tivoli Access Manager for WebLogic uses the Tivoli Access Manager Policy Server to browse protected objects in the Tivoli Access Manager protected object database. However, this architecture should only be used in test environments as the Tivoli Access Manager Policy Servers cannot be replicated and so introduces a Tivoli Access Manager for WebLogic single point of failure. Furthermore, the entitlements service boasts greater runtime performance based on internal caching techniques. The entitlements service architecture should always be used in production environments.

The following configuration steps should only be performed after Tivoli Access Manager for WebLogic has been correctly configured. Tivoli Access Manager for WebLogic makes use of two entitlements services that both need to be enabled on all configured Tivoli Access Manager Authorization servers:

- **Tivoli Access Manager Extended Attribute Entitlement Service**
  This is a default entitlement service that is distributed with the Tivoli Access Manager Authorization Server.

- **RBPF Protected Object Browsing Entitlement Service**
  This is an entitlement service that is distributed with Tivoli Access Manager for WebLogic.

To ensure Tivoli Access Manager for WebLogic is using the entitlements services, perform the following steps:

1. Copy the `rbpf_ent_pos_browser` shared library from the Tivoli Access Manager for WebLogic host to the Tivoli Access Manager Authorization Server host and place it in any directory that is located in the system PATH. The `rbpf_ent_pos_browser` shared library can be found on the Tivoli Access Manager for WebLogic host in:

   **UNIX**  
   `/opt/PolicyDirector/lib`

   **Windows**  
   `c:\Program Files\Tivoli\pdwls\bin`
2. From the Tivoli Access Manager Authorization host open the ivacld.conf file, located in:
   UNIX   /opt/PolicyDirector/etc
   Windows     c:\Program Files\Tivoli\Policy Director/etc

3. Add the following two lines to the [aznapi-entitlement-services] stanza:
   AZN_ENT_EXT_ATTR = azn_ent_ext_attr
   RBPF_POS_BROWSE = rbpf_ent_pos_browser


5. From the Tivoli Access Manager for WebLogic host, open the rbpf.properties file located in $java_home/amwls/WLS_Domain_Name/WLS_Realm_Name — where $WLS_Domain_Name is the name of the BEA WebLogic Server domain, and $WLS_Realm_Name is the name of the BEA WebLogic Server security realm.
   Update the following property to true:
   com.tivoli.pd.as.rbpf.UseEntitlements=true


Once these steps have been completed successfully the Tivoli Access Manager for WebLogic enabled BEA WebLogic Server will use the Tivoli Access Manager Authorization Server to perform all protected object browsing, as opposed to the Tivoli Access Manager Policy Server.

User and group management with Tivoli Access Manager for WebLogic

With Tivoli Access Manager for WebLogic, user and groups can be managed using the BEA WebLogic Server console. From the security pane of the BEA WebLogic Server console, expand the Access Manager icon then the Realm icon to display the Users and Groups icons. It is from these icons that you manage the users and groups for Tivoli Access Manager for WebLogic security.

Selecting the Users icon displays the User Management page. From this page you can:
- List Tivoli Access Manager for WebLogic users.
- Display the details of an individual user.
- Create users.

Selecting the Groups icon displays the Group Management page. From this page you can:
- List groups.
- Display the details of a particular group.
- Create groups.

You can add multiple users to groups, or groups to users, by entering a space separated list in the relevant console extension page.

When listing users or groups, if a value is not entered into the Max-Return field then all users or groups will be displayed that meet the criteria specified in the Pattern field.
Using the demonstration application

You can use the demonstration application to see an example of two types of authorization and to exercise the WebSEAL single sign-on capability.

The two types of authorization are:
- **Declarative**
  - Uses deployment descriptors to grant users and groups specific roles.
- **Programmatic**
  - Role checks are performed from within the application source code.

The demonstration application consists of a Web Component and an EJB component.

The two levels of security in the Web component can be described as follows:
- **Declarative:**
  - The web.xml deployment descriptor defines a single role named ServletRole. The weblogic.xml deployment descriptor defines a principal mapping between the ServletRole and a BankMembersServlet group. A security constraint in the web.xml deployment descriptor ensures users are required to be granted the ServletRole role to access any methods of the Servlet.
- **Programmatic:**
  - The doPost() method has additional security, this time, programmatically ensuring that the caller is granted the ServletRole. This allows testing of both programmatic and declarative security within a single web component. The HttpServletRequest.isUserInRole() method is used to make the authorization check.

The three levels of security in the EJB component can be described as follows:
- **Declarative security:**
  - A single role is defined in the ejb-jar.xml deployment descriptor named EJBRole. The weblogic-ejb-jar.xml deployment descriptor defines a principal mapping between EJBRole and a BankMembersEJB group. A method permission in the ejb-jar.xml deployment descriptor ensures users are required to be granted the EJBRole role in order to access the getBalance() method.
- **Programmatic security:**
  - The getBalance() method has further security, this time, programmatically ensuring the caller is granted the EJBRole. The EJBContext.isCallerInRole() method is used to make the authorization check.
- **Programmatic Security based on account name:**
  - The getBalance() method ensures that the name of the requested account matches the name of the calling principal. That is, only Banker1 should be able to view the account balance of Banker1.

To run the demonstration application, complete the following steps:
1. Copy the demonstration application P0DemoApp.ear into WebLogic_domain_directory\applications. Note that use of this directory is not required. You can place the EAR file into any directory on your file system. The demonstration application can be found in /AMWLS_install_dir/demo.
2. Use the BEA WebLogic Server console to create the following users:
   - Banker1
   - Banker2
   - Banker3
3. Create 2 groups: BankMembersEJB and BankMembersServlet. Add the users: Banker1, Banker2, Banker3 and Banker4, to the newly created groups. For instructions on using the BEA WebLogic Server console, see the BEA WebLogic Server documentation.

4. Use the BEA WebLogic Server console to deploy the demonstration application.

5. To access the demonstration application, access the following URL: http://WebLogic_Server_host:WebLogic_Server_listening_port/pddemo/PDDemo

Authenticate with one of the Banker users defined above.

WebLogic_Server_host is the hostname of the BEA WebLogic Server system.

WebLogic_Server_listening_port is the port on which the BEA WebLogic Server is listening.

6. Verify that only users who are in the BankMembersServlet group can access the servlet.

7. Verify that the authenticated users that are members of the BankMembersEJB group can view their own balances, but not the balance of any other user.

To test the WebSEAL single sign-on, complete the following steps:

1. Access the following URL:
   https://webseald_server_name/junction_target/pddemo/PDDemo

WebSEAL will prompt you to authenticate.

For an explanation of the variables webseald_server_name and junction_target, see “Part 7: Testing the configuration” on page 29.

Note: Use HTTPS here because the default WebSEAL behavior prevents Basic or Forms-based authentication over HTTP.

2. Authenticate as one of the users defined above.
   This process enables the user to single sign-on to the BEA WebLogic Server, and the Servlet will be invoked without requiring a second authentication. When accessed through WebSEAL, the PDDemo demonstration application will show identical behavior to that shown when accessing the BEA WebLogic Server directly.

3. Verify that the authenticated users can view their own balances, but not the balance of any other user.

Usage tips

1. Observe good security practices when enabling single sign-on for external users. Ensure that authentication is performed only by the WebSEAL server. To achieve this, disable access to the BEA WebLogic Server by internal users; that is, users who do not access BEA WebLogic Server using WebSEAL. This can be done using a network connection filter. A connection filter allows you to protect resources at the network level, rather than using roles to restrict access.

2. Be aware that both Tivoli Access Manager and WebLogic Server keep track of failed authentication attempts. Each product maintains a security configuration setting that specifies the maximum number of allowable failed attempts before
the user account is locked out. The user will be locked out by the lesser of the two settings. For example, if WebLogic Server is configured to allow five login failures but Tivoli Access Manager is configured to allow only three login failures, the user will be locked out after three login failures.

Three strikes logon policy

The three strikes logon policy, available for LDAP-based Tivoli Access Manager installations, enables you to prevent computer password attacks by specifying a maximum number of failed log on attempts and a penalty lockout time. The policy creates a condition where a user must wait a period of time before making more log on attempts that fail. For example, a policy could dictate 3 failed attempts followed by a 180 second penalty. This type of logon policy can prevent random computer-generated log on attempts occurring many times a second.

The three strikes logon policy requires the joint contribution of two pdadmin policy command settings:

- Maximum number of failed log on attempts
  ```
  policy set max-login-failures
  ```
- Penalty for exceeding failed log on attempt setting
  ```
  policy set disable-time-interval
  ```
  The penalty setting can include an account lockout time interval or a complete disabling of the account.

If a logon policy is set (as an example) for three failed attempts followed by specific lockout time penalty, a fourth attempt (correct or incorrect) will result in an error page that states the account is temporarily unavailable because of password policy.

The time interval is specified in seconds - the minimum recommended time interval is 60 seconds.

If the `disable-time-interval` policy is set to `disable`, the user is locked out of the account and the LDAP `account valid` attribute for this user is set to `no`. An administrator re-enables the account through the Web Portal Manager.

**Note:** Setting the `disable-time-interval` to `disable` results in additional administration overhead. You might observe delays in replicating `account valid` information to the plug-in. This situation depends on your LDAP environment. In addition, certain LDAP implementations might experience performance degradation as a result of the `account valid` update operation. For these reasons it is recommended that you use a timeout interval.

The following `pdadmin` commands are appropriate only for use with an LDAP registry.

**Table 3. pdadmin LDAP logon policy commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`policy set max-login-failures {number</td>
<td>unset} [-user username]`</td>
</tr>
<tr>
<td><code>policy get max-login-failures [-user username]</code></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Policy set disable-time-interval {number</td>
<td>unset</td>
</tr>
<tr>
<td>Policy get disable-time-interval [-user username]</td>
<td>Manages the policy controlling the maximum number of failed log on attempts allowed before a penalty is imposed. This command depends on a penalty set in the policy set disable-time-interval command. As the administrator, you can apply this policy to a specific user or apply the policy globally to all users listed in the LDAP registry. The default setting is 10 attempts.</td>
</tr>
</tbody>
</table>

**Deleting the Tivoli Access Manager Realm**

To delete the Tivoli Access Manager realm:

1. Ensure BEA WebLogic Server is started.
2. Using the console, change the default realm such that it is not the realm created by the Tivoli Access Manager for WebLogic `create_realm` action.
4. To delete the Tivoli Access Manager realm using the Console:
   a. Open the *Access Manager* icon from the BEA WebLogic Server navigation bar.
   b. Click on the *Realm* icon. The *Realm Configuration* page is displayed.
   c. Click *Delete*. The *Delete Realm Configuration* page is displayed.
   d. Click OK. The *Create Realm* page is displayed with empty fields.
5. To delete the Tivoli Access Manager realm using the command line, use the `AMWLSConfigure -action delete_realm`. Refer to Appendix B, “Command quick reference,” on page 53 for details on the options to use with the `AMWLSConfigure -action delete_realm` command.

**Note:** If Tivoli Access Manager for WebLogic was not installed into the recommended location during the file extraction, be sure to set the `AMSSPI_DIR` variable in the `AMWLSConfigure` script to the location of the actual installation directory. Similarly, if WebLogic is not installed in the default location update the `WLS_JAR` variable with the correct location of `WebLogic.jar` in the `ALWLSConfigure` script.
Unconfiguring Tivoli Access Manager for WebLogic

To unconfigure Tivoli Access Manager for WebLogic:
1. Ensure BEA WebLogic Server is started.
2. Ensure the Tivoli Access Manager realm has been deleted. Refer to “Deleting the Tivoli Access Manager Realm” on page 38.
3. To unconfigure Tivoli Access Manager for WebLogic using the console:
   a. Click the Access Manager folder. The Configuration page is displayed.
   b. Click Delete. The Unconfigure page is displayed.
   c. Enter the Tivoli Access Manager sec_master password and click OK.
   d. The Configuration page is displayed with empty fields.
4. To unconfigure Tivoli Access Manager for WebLogic from the command line use the AMWLSCONFIGURE -action unconfig command. Refer to Appendix B, “Command quick reference,” on page 53 for details on the options to use with the AMWLSCONFIGURE -action unconfig command.

Troubleshooting tips

Topic index:
- “Single sign-on failure using forms-based login”
- “WebLogic Server throws memory exception”

Single sign-on failure using forms-based login

When users have authenticated through forms-based login, and attempt to access a resource for which they do not have permission, the following error message may be displayed.

Could not Sign On message from WebSEAL

This can occur because even though the users could actually be authenticated, they do not have permission to access the Servlet in the Web container.

If this error occurs when using Basic Authentication, the users will be re-prompted for the authentication details, instead of seeing the page described above. This is default BEA WebLogic Server behavior and would be seen if the users access the page either directly or through WebSEAL.

WebLogic Server throws memory exception

Problem: A java.lang.OutOfMemory exception is thrown.

Explanation: When running a large number of Access Manager for WebLogic Server sessions, BEA WebLogic Server may run out of heap space.

Solution: Increase the maximum heap size option for the Java Virtual Machine (JVM) in the startWebLogic script. For example:

```
%JAVA_HOME%\bin\java -ms64m -mx128m -xms200m -xx:MaxPermSize=128m
```

Consult the BEA product documentation for recommended heap size, based on application architecture, the number of memory-intensive processes running on the host system, and the version of BEA WebLogic Server. Applications should be stress-tested to determine the appropriate heap size for their environment.
Limitations

1. Tivoli Access Manager for WebLogic does not support recursive group membership (groups within groups).

2. Tivoli Access Manager for WebLogic supports multiple Tivoli Access Manager domains, however, the sec_master user for each domain MUST be called sec_master. That is, there is not currently an option to change this user name for each Tivoli Access Manager domain.

3. In BEA WebLogic Server 8.1, use anyother instead of any-other as a group name as the "-" character is not supported for group names.

4. When configuring Tivoli Access Manager for WebLogic against Active Directory, you will need to change the AdminGroupProp=Administrators setting to something else, as the administrators group already exists in Active Directory and the configuration will fail. It is important that this is done prior to configuring Tivoli Access Manager for WebLogic and creating the Tivoli Access Manager for WebLogic realm.

5. When using Tivoli Access Manager for WebLogic console to create roles and polices, time restrictions are not supported. You cannot add users or groups to policy only roles. You can only use "OR" between roles and polices, "AND" is not supported.

6. Tivoli Access Manager caches user credentials for 2 hours by default. You can configure this time value by updating the appsrv-credcache-life property in PdPerm.properties.

7. Single sign-on from WebSEAL or Tivoli Access Manager Plug-in for Web Servers to the WebLogic Server Console Extension is not supported. This, however, is not a great issue as the WebLogic Server Console will not generally be available to users accessing from the Internet.

Known Issues and Work-arounds

1. Installations using an Active Directory user registry may encounter problems when deploying the certificate application. This problem is due to hard-coded role mappings for the Administrator group and the system user. In Active Directory both the Administrator’s group and system user are predefined, and cannot be removed. To remove these errors and ensure that the correct security is placed on the certificate application, edit the deployment descriptor of the certificate.war web application, remove these mappings, and add mappings that correspond to your actual Administrator group and system user.

2. A problem exists in BEA WebLogic Server version 8.1 that does not permit Tivoli Access Manager for WebLogic to perform policy updates from the console. The BEA WebLogic Server Change Request (CR) number for this issue is CR125113. Policy updates using the console are not supported until this problem is fixed in a BEA WebLogic Server 8.1 service pack.
Chapter 6. Removal instructions

This chapter describes how to remove IBM Tivoli Access Manager for WebLogic Server.

Complete the instructions in one of the following sections:
- “Removing from Solaris”
- “Removing from Windows”
- “Removing from AIX” on page 42
- “Removing from HP-UX” on page 42

Removing from Solaris

Before proceeding with the removal of Tivoli Access Manager for WebLogic, ensure you have deleted the Tivoli Access Manager realm and unconfigured Tivoli Access Manager for WebLogic. For details on doing these tasks refer to “Deleting the Tivoli Access Manager Realm” on page 38 and “Unconfiguring Tivoli Access Manager for WebLogic” on page 39.

Use `pkgrm` to remove the Tivoli Access Manager for WebLogic on Solaris. Complete the following instructions:
1. Log in as `root`.
2. To remove Tivoli Access Manager for WebLogic, enter the following command:
   
   ```
   # pkgrm PDWLS
   ```
   
   A prompt is displayed asking you to confirm the removal of the selected package. Enter the letter `y`.
3. A warning is displayed advising you that scripts will be executed with super user permissions during the process of removal. Enter the letter `y`.

   A status message lists each file as it is removed. After the postremove script runs, a status message indicates that the removal of the software package was successful. The `pkgrm` utility exits.

   Removal of the Tivoli Access Manager for WebLogic package is complete.

   If you want to remove the IBM Tivoli Access Manager Base prerequisites (Tivoli Access Manager Base runtime environment, Tivoli Access Manager Base Java runtime environment, and the optional Tivoli Access Manager application development kit) follow the instructions in the *IBM Tivoli Access Manager Base Installation Guide*.

Removing from Windows

Before proceeding with the removal of Tivoli Access Manager for WebLogic, ensure you have deleted the Tivoli Access Manager realm and unconfigured Tivoli Access Manager for WebLogic. For details on doing these tasks refer to “Deleting the Tivoli Access Manager Realm” on page 38 and “Unconfiguring Tivoli Access Manager for WebLogic” on page 39.
Use the Windows Add/Remove Programs icon interface to remove the Tivoli Access Manager for WebLogic files. Complete the following instructions:

1. Log in as a Windows user with administrator privilege.
2. Double-click the Add/Remove Programs icon.
4. Click Change/Remove.
   The Tivoli Access Manager for WebLogic files are removed.
   The Maintenance Complete dialog box appears.
5. Click OK.

Removal of Tivoli Access Manager for WebLogic is complete.

If you want to remove the IBM Tivoli Access Manager Base prerequisites (Tivoli Access Manager Base runtime environment, Tivoli Access Manager Base Java runtime environment, and the optional Tivoli Access Manager application development kit) follow the instructions in the IBM Tivoli Access Manager Base Installation Guide.

---

**Removing from AIX**

Before proceeding with the removal of Tivoli Access Manager for WebLogic, ensure you have deleted the Tivoli Access Manager realm and unconfigured Tivoli Access Manager for WebLogic. For details on doing these tasks refer to "Deleting the Tivoli Access Manager Realm" on page 38 and "Unconfiguring Tivoli Access Manager for WebLogic" on page 39.

Use the `installp` utility to remove the Tivoli Access Manager for WebLogic for AIX package.

If you want to remove the IBM Tivoli Access Manager Base prerequisites (Tivoli Access Manager Base runtime environment, Tivoli Access Manager Base Java runtime environment, and the optional Tivoli Access Manager application development kit) follow the instructions in the IBM Tivoli Access Manager Base Installation Guide.

---

**Removing from HP-UX**

Before proceeding with the removal of Tivoli Access Manager for WebLogic, ensure you have deleted the Tivoli Access Manager realm and unconfigured Tivoli Access Manager for WebLogic. For details on doing these tasks refer to "Deleting the Tivoli Access Manager Realm" on page 38 and "Unconfiguring Tivoli Access Manager for WebLogic" on page 39.

Use `swremove` to remove the Tivoli Access Manager for WebLogic files. Complete the following instructions:

1. Log in as root.
2. To remove Tivoli Access Manager for WebLogic, enter the following command:
   ```bash
   # swremove PDWLS
   ```

   A series of status messages appear. A status message appears indicating that the analysis phase has succeeded. The `swremove` utility removes the Tivoli Access Manager for WebLogic files from the hard disk.
When the removal is complete, the `swremove` utility exits.

Removal of Tivoli Access Manager for WebLogic on HP-UX is now complete.

If you want to remove the IBM Tivoli Access Manager Base prerequisites (Tivoli Access Manager Base runtime environment, Tivoli Access Manager Base Java runtime environment, and the optional Tivoli Access Manager application development kit) follow the instructions in the *IBM Tivoli Access Manager Base Installation Guide*. 
Appendix A. Properties Files Reference

Data entered when configuring Tivoli Access Manager for WebLogic and creating the realm is stored in properties files. These properties files can be used to change the behavior of Tivoli Access Manager for WebLogic.

The properties files exist on `java_home/amwls/wls_domain_name/wls_realm_name/`. Where `wls_domain_name` is the name of your configured BEA WebLogic Server domain, and `wls_realm_name` is the name of your configured BEA WebLogic Server realm within the domain.

There are three properties files:

- `amsspi.properties`
  Contains configuration properties for aspects of the SSPI specific to BEA WebLogic Server.
- `rbpf.properties`
  Contains configuration properties for Tivoli Access Manager for WebLogic, for example, cache settings, role properties, and Tivoli Access Manager protected object space container names.
- `amwlsjlog.properties`
  The parameters in this file control the logging and tracing for Tivoli Access Manager for WebLogic, including the amount of trace/messaging performed. Note that having trace activated can have an impact on the performance of Tivoli Access Manager for WebLogic. We recommend that trace is only activated when attempting to determine the cause of a problem.

The following sections contain descriptions of the parameters found in each of the properties files.

The marker, `***`, denotes properties that are not entered when configuring Tivoli Access Manager for WebLogic. These properties are set to the default values at the time of configuration. If you want these values set as something other than the default, you will need to change the value of the property in the corresponding `.in` file PRIOR to configuring and creating the realm. The `config` and `create_realm` actions use the values in the `.in` files to create ACLs and Tivoli Access Manager protected objects, so they cannot be changed after configuration or realm creation. Properties not marked with `***` in the following sections can easily be changed after configuration.

The `.in` files can be found in `/pdwls_install_dir/etc`.

**amsspi.properties**

This section lists and describes the properties found in the `amsspi.properties` file.

```
com.tivoli.amwls.sspi.config.DeployerGroupProp***
  The default value is Deployers. By default BEA WebLogic Server has 4 administration groups, this property allows users to change the name of the Deployers administration group to something other than Deployers.
```

```
com.tivoli.amwls.sspi.config.MonitorGroupProp***
  The default value is Monitors. By default BEA WebLogic Server has 4
```
administration groups, this property allows users to change the name of the Monitors administration group to something other than Monitors.

**com.tivoli.amwls.sspi.config.OperatorGroupProp***
The default value is Operators. By default BEA WebLogic Server has 4 administration groups, this allows users to change the name of the Operators admin group to something other than Operators.

**com.tivoli.amwls.sspi.config.AdminGroupProp***
The default value is Administrators. By default BEA WebLogic Server has 4 administration groups, this allows users to change the name of the Administrator administration group to something other than Administrators. This is an important property for systems using Active Directory as this property MUST be updated because Windows already has an administration group named Administrators.

**com.tivoli.amwls.sspi.Authentication.GroupRegistryDelete**
The default value is true. This property determines whether groups are deleted in the underlying directory when Tivoli Access Manager groups are deleted. This is the same as turning on/off the -registry flag when deleting a group using pdadmin.

**com.tivoli.amwls.sspi.Authentication.UserRegistryDelete**
The default value is true. This determines whether users are deleted in the underlying directory when Tivoli Access Manager users are deleted. This is the same as turning on/off the -registry flag when deleting a user using pdadmin.

**com.tivoli.amwls.sspi.Authentication.ssoEnabled**
The default is false. This enables/disables single sign-on from WebSEAL or Tivoli Access Manager Plug-in for Web Servers to BEA WebLogic Server.

**com.tivoli.amwls.sspi.Authentication.ssoTrustId**
The user used to establish a trust association with WebSEAL or Tivoli Access Manager Plug-in for Web Servers to perform the single sign-on.

**com.tivoli.amwls.sspi.Authentication.ssoPasswd_expiry**
The default is 120 (minutes). This property specifies the length of time in minutes the SSO trust ID’s authentication is cached, once this time completes, the SSO user is authenticated against Tivoli Access Manager at the next SSO attempt.

**com.tivoli.amwls.sspi.RoleMapper.EnableWebProgRolecheck**
The default value is true. This property enables or disables Web programmatic role checks. This enables administrators to turn off programmatic security for Web applications.

**com.tivoli.amwls.sspi.RoleMapper.EnableEjbProgRolecheck**
The default value is true. This property enables or disables EJB programmatic role checks. This enables administrators to turn off programmatic security for EJ Bs.

**com.tivoli.amwls.sspi.Authentication.GroupDNPrefix**
For LDAP, the default value is cn=. This property allows administrators to change the prefix when creating groups from the console extension.

**com.tivoli.amwls.sspi.Authentication.UserDNPrefix**
For LDAP, the default value is cn=. This property allows administrators to change the prefix when creating users from the console extension.
**rbpf.properties**

This section lists and describes the properties found in the rbpf.properties file.

**com.tivoli.pd.as.rbpf.ProductName**
The default value is PDWLS. This property is used in comments and descriptions when creating Tivoli Access Manager objects and ACLs.

**com.tivoli.pd.as.rbpf.RoleContainerName***
The default value is Roles. After configuration this property is changed to Roles/$WLS_Domain_Name/$WLS_Realm_Name. Where $WLS_Domain_Name is the name of the configured BEA WebLogic Server domain and $WLS_Realm_Name is the name of the configured BEA WebLogic Server realm.

**com.tivoli.pd.as.rbpf.ResourceContainerName***
The default is Resources. After configuration this property is changed to Resources/$WLS_Domain_Name/$WLS_Realm_Name. Where $WLS_Domain_Name is the name of the configured BEA WebLogic Server domain and $WLS_Realm_Name is the name of the configured BEA WebLogic Server realm.

**com.tivoli.pd.as.rbpf.PosRoot***
The default is WebAppServer. This property is the absolute root of the object space for all roles and resources in Tivoli Access Manager for WebLogic.

**com.tivoli.pd.as.rbpf.ProductId***
The default is WLS. This property is combined with the PosRoot value to form the root of the object space for all roles and resources.

**com.tivoli.pd.as.rbpf.AMActionGroup***
The default is WebAppServer. This property is the default name for the action group that is used to store the action to be checked by Tivoli Access Manager for WebLogic access decisions.

**com.tivoli.pd.as.rbpf.AMAction***
The default is i for invoke. This action is checked when Tivoli Access Manager for WebLogic performs access decisions, it will be added to the AMActionGroup.

**com.tivoli.pd.as.cache.EnableDynamicRoleCaching**
The default is true. This property enables or disabled dynamic role caching. The dynamic role cache is used to cache all normal roles, that is, roles other than Administration roles. It caches positive and negative role memberships.

**com.tivoli.pd.as.cache.DynamicRoleCache**
The default is com.tivoli.pd.as.cache.DynamicRoleCacheImpl. This property is the class used to perform dynamic role caching. You can implement your own dynamic role cache if required. This can be done by implementing the com.tivoli.pd.as.cache.IDynamicRoleCache interface.

**com.tivoli.pd.as.cache.DynamicRoleCache.NumBuckets**
The default is 20. This property specifies the number of buckets that should be used in the underlying hash table that is used to store dynamic role cache entries.

**com.tivoli.pd.as.cache.DynamicRoleCache.MaxUsers**
The default is 100000. This property is the total number of entries for all buckets in the cache. This figure, divided by NumBuckets determines the max size of each individual bucket.
com.tivoli.pd.as.cache.DynamicRoleCache.RoleLifetime
The default is 20. This property specifies the length of time in seconds that positive and negative dynamic role cache decisions remain in the cache.

com.tivoli.pd.as.cache.DynamicRoleCache.PrincipalLifeTime
The default is 10. This property specifies the length of time in minutes that principal credentials are stored in the Tivoli Access Manager for WebLogic cache. Note, the PdPerm.properties value, appsr-credcache-life, determines how long credentials are cached in the PDJRTE. Tivoli Access Manager for WebLogic obtains all credentials from PDJRTE, thus, if this value is less than appsr-credcache-life it is overridden as Tivoli Access Manager for WebLogic retrieves a cached credential from PDJRTE.

com.tivoli.pd.as.cache.EnableStaticRoleCaching
The default is true. This property enables or disables static role caching. The static role cache is used to cache positive and negative role memberships for Administration roles. This cache is the same as the dynamic role cache except that entries do not expire. This enhances performance of administration roles as memberships to these roles are not expected to change.

com.tivoli.pd.as.cache.StaticRoleCache
The default is com.tivoli.pd.as.cache.StaticRoleCacheImpl. This is the class used to perform static role caching. You can implement your own static role cache if required. This can be done by implementing the com.tivoli.pd.as.cache.IStaticRoleCache interface.

com.tivoli.pd.as.cache.StaticRoleCache.Roles
The default is Admin, Operator, Monitor, Deployer. The property holds a comma separated list of Administrative roles. Memberships of roles in this list are added to the static role cache, rather than the dynamic role cache. All other role memberships are cached in the dynamic role cache.

com.tivoli.pd.as.cache.EnableObjectCaching
The default is true. This property enables or disables object caching. The object cache is used to cache all Tivoli Access Manager objects, including their extended attributes. This allows the caching of which roles are granted access to which BEA WebLogic Server resources; therefore bypassing the need to query the Tivoli Access Manager Authorization server for each resource request.

com.tivoli.pd.as.cache.ObjectCache
The default is com.tivoli.pd.as.cache.ObjectCacheImpl. This property is the class used to perform object caching. You can implement your own object cache if required. This can be done by implementing the com.tivoli.pd.as.cache.IObjectCache interface.

com.tivoli.pd.as.cache.ObjectCache.NumBuckets
The default is 20. This property specifies the number of buckets used to store object cache entries in the underlying hash table.

com.tivoli.pd.as.cache.ObjectCache.MaxResources
The default is 10000. This property specifies the total number of entries for all buckets in the cache. This figure, divided by NumBuckets determines the maximum size of each bucket.

com.tivoli.pd.as.cache.ObjectCache.ResourceLifeTime
The default is 20. This property specifies the length of time in minutes that objects are kept in the object cache.
com.tivoli.pd.as.rbpf.UncheckedRoles
The default is Unchecked, AmasUnchecked, Anonymous. This property specifies a comma separated list of J2EE unchecked roles. If any of the listed roles are granted access to a BEA WebLogic Server resource, all users are granted access to it, regardless of which normal roles are attached. Users and groups cannot be added to these roles. These roles represent an efficient way to grant access for all users (including unauthenticated) to a specific resource. The Anonymous role should always remain in this list as the Tivoli Access Manager for WebLogic configuration adds this unchecked role to several base BEA WebLogic Server resources. This property does not have to be set before configuration, however once set it should not be changed.

com.tivoli.pd.as.rbpf.ExcludedRoles
The default is Excluded, AmasExcluded. This property specifies a comma separated list of J2EE excluded roles. Thus, if any of these roles are attached to a resource no users will be granted access to it, regardless of which normal roles are attached. These J2EE excluded roles represent an efficient way to deny access for all users to a specific resource. This property does not have to be set before configuration, however once set it should not be changed.

com.tivoli.pd.as.rbpf.GrantUnprotectedAccess
The default is true. This property specifies whether to grant or deny access to a requested resource that is unprotected; that is, an object that does not have any roles granted to it.

com.tivoli.pd.as.rbpf.CopyParentRole***
The default is false. This property allows administrators to specify whether role members defined at a higher level (for example, a global role) should be copied when creating a role at a more specific level (for example, a role at the application level). In Tivoli Access Manager, this involves copying all members of an ACL attached at the global level to the ACL attached to the object at the application level. This property gives administrators the power to apply the concept of inheritance to role memberships when creating a new role. Generally this should be set to the same value as PropogateChildRole.

com.tivoli.pd.as.rbpf.PropagateChildRole***
The default is false. This property allows administrators to specify whether changes made to role memberships defined at a higher level (for example, a global role) are also made to child roles (for example, roles at the application level). This is, adding userA to a global role, RoleA, would also add userA to RoleA at the application level. This enhances the CopyParentRole and further applies role member inheritance when updating role memberships. Generally, this property should be set to the same value as CopyParentRole.

com.tivoli.pd.as.rbpf.UseEntitlements
The default is false. This property indicates whether an entitlements service in the Tivoli Access Manager Authorization Server should be used to gather information about which roles are granted access to which resources. This defaults to false so you are able to set up the minimum number of Tivoli Access Manager services to get Tivoli Access Manager for WebLogic running. However, this property should only be set to false in test environments as it has a single point of failure against the Tivoli Access Manager Policy Server. The entitlements service also performs at a
much higher level based on internal object caching. Thus, in production environments this value should always be set to true.

**com.tivoli.pd.as.rbpf.EntitlementsUser**
The default is the Tivoli Access Manager for WebLogic remote-acl-user. This property holds the user employed to perform object lookups using the entitlements service. The entitlements service ensures the user requesting objects from the Tivoli Access Manager protected object space is granted the Server Admin Generic ‘s’ permission. During **config** the remote-acl-user is added to the iv-admin group and granted this permission. The user requesting objects can be updated by changing this user, however, you need to ensure that this new user is granted the ‘s’ permission on the Resources container in the Tivoli Access Manager protected object space.

**com.tivoli.pd.as.rbpf.IgnorePasswordPolicyOnUserCreate**
The default is **false**. This property allows administrators to ignore password policy when creating new Tivoli Access Manager users through the BEA WebLogic Server console.

**com.tivoli.pd.as.rbpf.DeleteBaseRoleRecursive**
The default is **true**. This property indicates whether to delete all child roles when deleting a parent role.

### amwlsjlog.properties

The **amwlsjlog.properties** file is a standard JLog properties file. It is used to control messaging and tracing in Tivoli Access Manager for WebLogic as well as PDJRTE.

This section does not list all the properties contained in the **amwlsjlog.properties** file as most are irrelevant for our purposes. It is from this file, though, that messaging and tracing is enabled or disabled.

The entries in the **amwlsjlog.properties** file are hierarchical in nature. You can turn logging on for several components at once, or you can turn it on for a single component.

To turn on logging you simply add an **isLogging** property to the component for which you want to enable logging. Listed below are all the tracing and messaging components supported by Tivoli Access Manager for WebLogic. You can enable trace/messaging for 1 of these listed properties, or for all of them. Following is a brief description of what each component does.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tracing</strong></td>
<td></td>
</tr>
<tr>
<td><strong>AmasRBPFTraceLogger</strong></td>
<td>Trace for internal operations of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td><strong>AmasCacheTraceLogger</strong></td>
<td>Trace for all Tivoli Access Manager for WebLogic caches.</td>
</tr>
<tr>
<td><strong>AMSSPICfgTraceLogger</strong></td>
<td>Trace for the <strong>config</strong> operations of Tivoli Access Manager for WebLogic,</td>
</tr>
<tr>
<td></td>
<td>for example, Role creation.</td>
</tr>
<tr>
<td><strong>AMSSPIAuthzTraceLogger</strong></td>
<td>Trace for the authorization provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AMSSPIAuthnTraceLogger</td>
<td>Trace for the authentication provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AMSSPIRoleMapperTraceLogger</td>
<td>Trace for the role mapping provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AMSSPIResourceManagerTraceLogger</td>
<td>Trace for the resource manager within Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td><strong>Messaging</strong></td>
<td></td>
</tr>
<tr>
<td>AmasCacheMessageLogger</td>
<td>Messaging for the internal operations of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AmasRBPFMessageLogger</td>
<td>Messaging for all Tivoli Access Manager for WebLogic caches.</td>
</tr>
<tr>
<td>AMSSPICfgMessageLogger</td>
<td>Messaging for the config operations of Tivoli Access Manager for WebLogic, for example, Role creation.</td>
</tr>
<tr>
<td>AMSSPIAuthzMessageLogger</td>
<td>Messaging for the authorization provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AMSSPIAuthnMessageLogger</td>
<td>Messaging for the authentication provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AMSSPIRoleMapperMessageLogger</td>
<td>Messaging for the role mapping provider of Tivoli Access Manager for WebLogic.</td>
</tr>
<tr>
<td>AMSSPIResourceManagerMessageLogger</td>
<td>Messaging for the resource manager within Tivoli Access Manager for WebLogic.</td>
</tr>
</tbody>
</table>

Each of the above components extend the **baseGroup traceLogger** and **baseGroup messageLogger**. Thus, in the properties file their properties will appear similar to this example:

```
baseGroup.AMSSPIAuthnMessageLogger.isLogging=true
```

The above example enables messaging for the Authentication provider section of Tivoli Access Manager for WebLogic. If you want to enable tracing for all components, except for the authorization provider, you would add the following lines:

```
baseGroup.TraceLogger.isLogging=true
baseGroup.AMSSPIAuthzMessageLogger.isLogging=false
```

That is, all other tracing components would simply inherit the `true` value from the base logger. Where as the authorization logger overrides the `true` value with `false`. 
Appendix B. Command quick reference
AMWLSConfigure –action config

Configures Tivoli Access Manager for WebLogic Server.

Syntax

AMWLSConfigure –action config –domain_admin domain_admin
–domain_admin_pwd domain_admin_password –remote_acl_user remote_acl_user
–sec_master_pwd sec_master_pwd –pdmgrd_host pdmgrd_host –pdacld_host pdacld_host
[pdagld_host [–deploy_extension true|false] [–wls_server_url wls_server_url]
[–am_domain am_domain] [–pdmgrd_port pdmgrd_port] [–pdacld_port pdacld_port]
[–amwls_home amwls_home] [–verbose true|false]]

Parameters

–am_domain am_domain
  Specifies the name of the Tivoli Access Manager domain. The default domain
  is Default.

–amwls_home amwls_home
  Specifies the path to the Tivoli Access Manager for WebLogic Server
  installation directory.

–deploy_extension true|false
  Deploys the Tivoli Access Manager Web Logic Server version 5.1 console
  extension when set to true. The default value is true.

–domain_admin domain_admin
  Specifies the WebLogic domain administrator.

–domain_admin_pwd domain_admin_password
  Specifies the WebLogic domain administrator password.

–pdacld_host pdacld_host
  Specifies the Tivoli Access Manager authorization server host name.

–pdacld_port pdacld_port
  Specifies the Tivoli Access Manager authorization server port number. The
  default port number is 7136.

–pdmgrd_host pdmgrd_host
  Specifies the Tivoli Access Manager policy server host name.

–pdmgrd_port pdmgrd_port
  Specifies the Tivoli Access Manager policy server port number. The default
  port number is 7135.

–remote_acl_user remote_acl_user
  Specifies the Tivoli Access Manager principal that is created for the
  authorization server.

–sec_master_pwd sec_master_pwd
  Specifies the Tivoli Access Manager administrative user password (normally
  sec_master).

–verbose true|false
  Enables verbose output when set to true. The default value is false.

–wls_server_url wls_server_url
  Specifies the URL for the local WebLogic Server. The default is
  t3://localhost:7001
Availability

This command is located in the following default installation directories:

- UNIX systems:
  /opt/pdwls/sbin/
- On Windows systems:
  C:\Program Files\Tivoli\pdwls\sbin\

When an installation directory other than the default is selected, this utility is located in the sbin directory under the installation directory (for example, install_dir\sbin\).

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.

When the command fails, an error message is displayed. Refer to the IBM Tivoli Access Manager Error Message Reference for a more detailed description of the problem.
AMWLSConfigure –action unconfig

Unconfigures Tivoli Access Manager for WebLogic Server.

Syntax

AMWLSConfigure –action unconfig –domain_admin_pwd domain_admin_pwd
–sec_master_pwd sec_master_pwd [–verbose {true | false}]

Parameters

–domain_admin_pwd domain_admin_pwd
  Specifies the Tivoli Access Manager for WebLogic Server domain administrator password.

–sec_master_pwd sec_master_pwd
  Specifies the Tivoli Access Manager administrative user password (usually sec_master).

–verbose {true | false}
  Enables verbose output when set to true. The default value is false.

Availability

This command is located in the following default installation directories:

- UNIX systems:
  /opt/pdwls/sbin/

- On Windows systems:
  C:\Program Files\Tivoli\pdwls\sbin\

When an installation directory other than the default is selected, this utility is located in the sbin directory under the installation directory (for example, install_dir\sbin\).

Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.

When the command fails, an error message is displayed. Refer to the IBM Tivoli Access Manager Error Message Reference for a more detailed description of the problem.
AMWLSConfigure –action create_realm

Creates the security realm in WebLogic Server.

Syntax

AMWLSConfigure –action create_realm  
–realm_name realm_name
–domain_admin_pwd domain_admin_pwd  
–group_dn_suffix user_dn_suffix
–group_dn_prefix group_dn_prefix
–admin_group admin_group  
–user_dn_prefix user_dn_prefix
–sso_enabled {true|false}
–sso_user sso_user
–sso_pwd sso_pwd
–verbose {true|false}

Parameters

–admin_group admin_group  
Specifies the Tivoli Access Manager group to use for internal configuration purposes.

–domain_admin_pwd domain_admin_pwd  
Specifies the WebLogic domain administrator password.

–group_dn_prefix group_dn_prefix  
Specifies the distinguished name (DN) prefix to use when creating groups.

–group_dn_suffix group_dn_suffix
–realm_name realm_name  
Specifies the distinguished name (DN) suffix to use when creating groups.

–realm_name realm_name
–sso_enabled {true|false}
–sso_user sso_user
–sso_pwd sso_pwd
–verbose {true|false}  
Enables single signon support when set to true. The default value is false.

–user_dn_prefix user_dn_prefix
–user_dn_suffix user_dn_suffix
–admin_group admin_group

Availability

This command is located in the following default installation directories:

- UNIX systems:
  /opt/pdwls/sbin/

- On Windows systems:
  C:\Program Files\Tivoli\pdwls\sbin

When an installation directory other than the default is selected, this utility is located in the sbin directory under the installation directory (for example, install_dir\sbin).
Return Codes

The following exit status codes can be returned:

0  The command completed successfully.
1  The command failed.

When the command fails, an error message is displayed. Refer to the *IBM Tivoli Access Manager Error Message Reference* for a more detailed description of the problem.
**AMWLSConfigure –action delete_realm**

Deletes the security realm from WebLogic Server.

**Syntax**

\[ AMWLSConfigure \ –action \ delete_realm \ –domain_admin_pwd \ domain_admin_pwd \ [-registry_clean \ {true|false}] \ [-verbose \ {true|false}] \]

**Parameters**

- **–domain_admin_pwd \ domain_admin_pwd**
  Specifies the WebLogic domain administrator password.

- **–registry_clean \ {true|false}**
  Removes the users and groups that were created during configuration. The default value is false.

- **–verbose \ {true|false}**
  Enables verbose output when set to true. The default value is false.

**Availability**

This command is located in the following default installation directories:

- UNIX systems:
  /opt/pdwls/sbin/

- On Windows systems:
  C:\Program Files\Tivoli\pdwls\sbin\n
When an installation directory other than the default is selected, this utility is located in the sbin directory under the installation directory (for example, \install_dir\sbin\).

**Return Codes**

The following exit status codes can be returned:

0  The command completed successfully.

1  The command failed.

When the command fails, an error message is displayed. Refer to the *IBM Tivoli Access Manager Error Message Reference* for a more detailed description of the problem.
Appendix C. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
DB2
IBM
IBM logo
SecureWay
Tivoli
Tivoli logo

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-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 of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
Glossary

A

access control. In computer security, the process of ensuring that the resources of a computer system can be accessed only by authorized users in authorized ways.

access control list (ACL). In computer security, a list that is associated with an object that identifies all the subjects that can access the object and their access rights. For example, an access control list is a list that is associated with a file that identifies the users who can access the file and identifies the users’ access rights to that file.

access permission. The access privilege that applies to the entire object.

action. An access control list (ACL) permission attribute. See also access control list.

ACL. See access control list.

administration service. An authorization API runtime plug-in that can be used to perform administration requests on a Tivoli Access Manager resource management application. The administration service will respond to remote requests from the pdadmin command to perform tasks, such as listing the objects under a particular node in the protected object tree. Customers may develop these services using the authorization ADK.

attribute list. A linked list that contains extended information that is used to make authorization decisions. Attribute lists consist of a set of name = value pairs.

authentication. (1) In computer security, verification of the identity of a user or the user’s eligibility to access an object. (2) In computer security, verification that a message has not been altered or corrupted. (3) In computer security, a process that is used to verify the user of an information system or of protected resources. See also multi-factor authentication, network-based authentication, and step-up authentication.

authorization. (1) In computer security, the right granted to a user to communicate with or make use of a computer system. (2) The process of granting a user either complete or restricted access to an object, resource, or function.

authorization rule. See rule.

authorization service plug-in. A dynamically loadable library (DLL or shared library) that can be loaded by the Tivoli Access Manager authorization API runtime client at initialization time in order to perform operations that extend a service interface within the Authorization API. The service interfaces that are currently available include Administration, External Authorization, Credentials modification, Entitlements and PAC manipulation interfaces. Customers may develop these services using the authorization ADK.

B

BA. See basic authentication.

basic authentication. A method of authentication that requires the user to enter a valid user name and password before access to a secure online resource is granted.

bind. To relate an identifier to another object in a program; for example, to relate an identifier to a value, an address or another identifier, or to associate formal parameters and actual parameters.

blade. A component that provides application-specific services and components.

business entitlement. The supplemental attribute of a user credential that describes the fine-grained conditions that can be used in the authorization of requests for resources.

C

CA. See certificate authority.

CDAS. See Cross Domain Authentication Service.

CDMF. See Cross Domain Mapping Framework.

certificate. In computer security, a digital document that binds a public key to the identity of the certificate owner, thereby enabling the certificate owner to be authenticated. A certificate is issued by a certificate authority.

certificate authority (CA). An organization that issues certificates. The certificate authority authenticates the certificate owner’s identity and the services that the owner is authorized to use, issues new certificates, renews existing certificates, and revokes certificates belonging to users who are no longer authorized to use them.

cGI. See common gateway interface.
ciphertext. Encrypted data that is unreadable until it has been converted into plain data (decrypted) with a key.

cross domain authentication service (CDAS). A WebSEAL service that provides a shared library mechanism that allows you to substitute the default WebSEAL authentication mechanisms with a custom process that returns a Tivoli Access Manager identity to WebSEAL. See also WebSEAL.

cross domain mapping framework (CDMF). A programming interface that allows a developer to customize the mapping of user identities and the handling of user attributes when WebSEAL e-Community SSO function are used.

d daemon. A program that runs unattended to perform continuous or periodic systemwide functions, such as network control. Some daemons are triggered automatically to perform their task; others operate periodically.

directory schema. The valid attribute types and object classes that can appear in a directory. The attribute types and object classes define the syntax of the attribute values, which attributes must be present, and which attributes may be present for the directory.

distinguished name (DN). The name that uniquely identifies an entry in a directory. A distinguished name is made up of attribute=value pairs, separated by commas.

domain. (1) A logical grouping of users, systems, and resources that share common services and usually function with a common purpose. (2) That part of a computer network in which the data processing resources are under common control. See also domain name.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames that are separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of a host system is as400.rchland.vnet.ibm.com, each of the following is a domain name: as400.rchland.vnet.ibm.com, vnet.ibm.com, ibm.com.

d EAS. See External Authorization Service.

d encryption. In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

d entitlement. A data structure that contains externalized security policy information. Entitlements contain policy data or capabilities that are formatted in a way that is understandable to a specific application.

d entitlement service. An authorization API runtime plug-in which can be used to return entitlements from an external source for a principal or set of conditions. Entitlements are normally application specific data that will be consumed by the resource manager application
in some way or added to the principal’s credentials for use further on in the authorization process. Customers may develop these services using the authorization ADK.

**external authorization service.** An authorization API runtime plug-in that can be used to make application or environment specific authorization decisions as part of the Tivoli Access Manager authorization decision chain. Customers may develop these services using the authorization ADK.

**F**

**file transfer protocol (FTP).** In the Internet suite of protocols, an application layer protocol that uses Transmission Control Protocol (TCP) and Telnet services to transfer bulk-data files between machines or hosts.

**G**

**global signon (GSO).** A flexible single sign-on solution that enables the user to provide alternative user names and passwords to the back-end Web application server. Global signon grants users access to the computing resources they are authorized to use — through a single login. Designed for large enterprises consisting of multiple systems and applications within heterogeneous, distributed computing environments, GSO eliminates the need for users to manage multiple user names and passwords. See also single signon.

GSO. See global signon.

**H**

**host.** A computer that is connected to a network (such as the Internet or an SNA network) and provides an access point to that network. Also, depending on the environment, the host may provide centralized control of the network. The host can be a client, a server, or both a client and a server simultaneously.

**HTTP.** See Hypertext Transfer Protocol.

**hypertext transfer protocol (HTTP).** In the Internet suite of protocols, the protocol that is used to transfer and display hypertext documents.

**I**

**Internet protocol (IP).** In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network.

**Internet suite of protocols.** A set of protocols developed for use on the Internet and published as Requests for Comments (RFCs) through the Internet Engineering Task Force (IETF).

**interprocess communication (IPC).** (1) The process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of interprocess communication. (2) A mechanism of an operating system that allows processes to communicate with each other within the same computer or over a network.

**IP.** See Internet Protocol.

**IPC.** See Interprocess Communication.

**J**

**junction.** An HTTP or HTTPS connection between a front-end WebSEAL server and a back-end Web application server. WebSEAL uses a junction to provide protective services on behalf of the back-end server.

**K**

**key.** In computer security, a sequence of symbols that is used with a cryptographic algorithm for encrypting or decrypting data. See private key and public key.

**key database file.** See key ring.

**key file.** See key ring.

**key pair.** In computer security, a public key and a private key. When the key pair is used for encryption, the sender uses the public key to encrypt the message, and the recipient uses the private key to decrypt the message. When the key pair is used for signing, the signer uses the private key to encrypt a representation of the message, and the recipient uses the public key to decrypt the representation of the message for signature verification.

**key ring.** In computer security, a file that contains public keys, private keys, trusted roots, and certificates.

**L**

**LDAP.** See Lightweight Directory Access Protocol.

**lightweight directory access protocol (LDAP).** An open protocol that (a) uses TCP/IP to provide access to directories that support an X.500 model and (b) does not incur the resource requirements of the more complex X.500 Directory Access Protocol (DAP). Applications that use LDAP (known as directory-enabled applications) can use the directory as a common data store and for retrieving information about people or services, such as e-mail addresses, public keys, or service-specific configuration parameters. LDAP was originally specified in RFC
1777. LDAP version 3 is specified in RFC 2251, and the IETF works on additional standard functions. Some of the IETF-defined standard schemas for LDAP are found in RFC 2256.

**lightweight third party authentication (LTPA).** An authentication framework that allows single sign-on across a set of Web servers that fall within an Internet domain.

**LTPA.** See lightweight third party authentication.

**M**

**management domain.** The default domain in which Tivoli Access Manager enforces security policies for authentication, authorization, and access control. This domain is created when the policy server is configured. See also domain.

**management server.** Obsolete. See policy server.

**metadata.** Data that describes the characteristics of stored data.

**migration.** The installation of a new version or release of a program to replace an earlier version or release.

**multi-factor authentication.** A protected object policy (POP) that forces a user to authenticate using two or more levels of authentication. For example, the access control on a protected resource can require that the users authenticate with both user name/password and user name/token passcode. See also protected object policy.

**multiplexing proxy agent (MPA).** A gateway that accommodates multiple client access. These gateways are sometimes known as Wireless Access Protocol (WAP) gateways when clients access a secure domain using a WAP. Gateways establish a single authenticated channel to the originating server and tunnel all client requests and responses through this channel.

**N**

**network-based authentication.** A protected object policy (POP) that controls access to objects based on the internet protocol (IP) address of the user. See also protected object policy.

**P**

**PAC.** See privilege attribute certificate.

**permission.** The ability to access a protected object, such as a file or directory. The number and meaning of permissions for an object are defined by the access control list (ACL). See also access control list.

**policy.** A set of rules that are applied to managed resources.

**policy server.** The Tivoli Access Manager server that maintains the location information about other servers in the secure domain.

**polling.** The process by which databases are interrogated at regular intervals to determine if data needs to be transmitted.

**POP.** See protected object policy.

**portal.** An integrated Web site that dynamically produces a customized list of Web resources, such as links, content, or services, available to a specific user, based on the access permissions for the particular user.

**privilege attribute certificate.** A digital document that contains a principal’s authentication and authorization attributes and a principal’s capabilities.

**privilege attribute certificate service.** An authorization API runtime client plug-in which translates a PAC of a predetermined format into a Tivoli Access Manager credential, and vice-versa. These services could also be used to package or marshall a Tivoli Access Manager credential for transmission to other members of the secure domain. Customers may develop these services using the authorization ADK. See also privilege attribute certificate.

**protected object.** The logical representation of an actual system resource that is used for applying ACLs and POPs and for authorizing user access. See also protected object policy and protected object space.

**protected object policy (POP).** A type of security policy that imposes additional conditions on the operation permitted by the ACL policy to access a protected object. It is the responsibility of the resource manager to enforce the POP conditions. See also access control list, protected object, and protected object space.

**protected object space.** The virtual object representation of actual system resources that is used for applying ACLs and POPs and for authorizing user access. See also protected object and protected object policy.

**private key.** In computer security, a key that is known only to its owner. Contrast with public key.

**public key.** In computer security, a key that is made available to everyone. Contrast with private key.

**Q**

**quality of protection.** The level of data security, determined by a combination of authentication, integrity, and privacy conditions.
**R**

**registry.** The datastore that contains access and configuration information for users, systems, and software.

**replica.** A server that contains a copy of the directory or directories of another server. Replicas back up servers in order to enhance performance or response times and to ensure data integrity.

**resource object.** The representation of an actual network resource, such as a service, file, and program.

**response file.** A file that contains a set of predefined answers to questions asked by a program and that is used instead of entering those values one at a time.

**role activation.** The process of applying the access permissions to a role.

**role assignment.** The process of assigning a role to a user, such that the user has the appropriate access permissions for the object defined for that role.

**routing file.** An ASCII file that contains commands that control the configuration of messages.

**RSA encryption.** A system for public-key cryptography used for encryption and authentication. It was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The system’s security depends on the difficulty of factoring the product of two large prime numbers.

**rule.** One or more logical statements that enable the event server to recognize relationships among events (event correlation) and to execute automated responses accordingly.

**run time.** The time period during which a computer program is executing. A runtime environment is an execution environment.

**S**

**scalability.** The ability of a network system to respond to increasing numbers of users who access resources.

**schema.** The set of statements, expressed in a data definition language, that completely describe the structure of a database. In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

**secure sockets layer (SSL).** A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

**security management.** The management discipline that addresses an organization’s ability to control access to applications and data that are critical to its success.

**self-registration.** The process by which a user can enter required data and become a registered Tivoli Access Manager user, without the involvement of an administrator.

**service.** Work performed by a server. A service can be a simple request for data to be sent or stored (as with file servers, HTTP servers, e-mail servers, and finger servers), or it can be more complex work such as that of print servers or process servers.

**silent installation.** An installation that does not send messages to the console but instead stores messages and errors in log files. Also, a silent installation can use response files for data input. See also response file.

**single signon (SSO).** The ability of a user to logon once and access multiple applications without having to logon to each application separately. See also global signon.

**SSL.** See Secure Sockets Layer.

**SSO.** See Single Signon.

**step-up authentication.** A protected object policy (POP) that relies on a preconfigured hierarchy of authentication levels and enforces a specific level of authentication according to the policy set on a resource. The step-up authentication POP does not force the user to authenticate using multiple levels of authentication to access any given resource but requires the user to authenticate at a level at least as high as that required by the policy protecting a resource.

**suffix.** A distinguished name that identifies the top entry in a locally held directory hierarchy. Because of the relative naming scheme used in Lightweight Directory Access Protocol (LDAP), this suffix applies to every other entry within that directory hierarchy. A directory server can have multiple suffixes, each identifying a locally held directory hierarchy.

**T**

**token.** (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (2) In local area networks (LANs), a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.
trusted root. In the Secure Sockets Layer (SSL), the public key and associated distinguished name of a certificate authority (CA).

U

uniform resource identifier (URI). The character string used to identify content on the Internet, including the name of the resource (a directory and file name), the location of the resource (the computer where the directory and file name exist), and how the resource can be accessed (the protocol, such as HTTP). An example of a URI is a uniform resource locator, or URL.

uniform resource locator (URL). A sequence of characters that represent information resources on a computer or in a network such as the Internet. This sequence of characters includes (a) the abbreviated name of the protocol used to access the information resource and (b) the information used by the protocol to locate the information resource. For example, in the context of the Internet, these are abbreviated names of some protocols used to access various information resources: http, ftp, gopher, telnet, and news; and this is the URL for the IBM home page: http://www.ibm.com.

URI. See uniform resource identifier.

URL. See uniform resource locator.

user. Any person, organization, process, device, program, protocol, or system that uses a service provided by others.

user registry. See registry.

V

virtual hosting. The capability of a Web server that allows it to appear as more than one host to the Internet.

W

Web Portal Manager (WPM). A Web-based graphical application used to manage Tivoli Access Manager Base and WebSEAL security policy in a secure domain. An alternative to the pdadmin command line interface, this GUI enables remote administrator access and enables administrators to create delegated user domains and assign delegate administrators to these domains.

WebSEAL. A Tivoli Access Manager blade. WebSEAL is a high performance, multi-threaded Web server that applies a security policy to a protected object space. WebSEAL can provide single sign-on solutions and incorporate back-end Web application server resources into its security policy.
Index

A
Access Manager
    Java runtime environment 11, 19
    pdjrtecfg 19
    policy server 10
    security model 1
    WebSEAL 10
AIX
    installing 14
    removing from 42
AMWLSConfigure -action config 54
AMWLSConfigure -action create_realm 57
AMWLSConfigure -action delete realm 59
AMWLSConfigure -action unconfig 56
authentication
    Access Manager 5
    of external users 5
    using WebSEAL 5
    without WebSEAL 6
authorization
    declarative 35
    programmatic 35

B
basic authentication
    configured user 5

C
CLASSPATH
    setting for startWebLogic 20
    setting for startWebLogic with language pack 21
configured user 5, 36
creating
    WebSEAL junction
        using pdadmin 27

D
declarative authorization 35
demonstration application 35
disk requirements 9

H
HP-UX
    installing on 15
    removing from 42

I
installation 14
    on AIX 14
    on HP-UX 15
    on Solaris 15
    on Windows 16
installp 14

J
Java
    runtime on AIX 11
    junctions
        configuring 27

L
language pack
    non-English 21
limitations
    group within group 40
    java.security.ACL interface 40
    managing J2EE resources 40
    logon policy 37

M
memory requirements 9

P
pdadmin
    creating a WebSEAL junction 27
    pdjrtecfg
        command line 19
    pkgadd 15
    pkgrm 41
    policy
        logon 37
    policy server 10
    prerequisites
        software 10
    problem determination 39
    programmatic authorization 35

R
related publications viii
removal instructions
    AIX 42
    HP-UX 42
    Solaris 41
    Windows 41
removing Tivoli Access Manager for WebLogic
    how to 41

S
single sign-on 10
    testing with demonstration application 36
SMIT 42
Solaris
    installing on 15
    removing from 41
startWebLogic
    command location 21
startWebLogic, Setting CLASSPATH for 20
supported platforms 9
swinstall 15
swremove 42

T
troubleshooting
  authentication 39
  out of memory problem 39

U
usage tips 36
utilities
  AMWLSConfigure -action config 54
  AMWLSConfigure -action create_realm 57
  AMWLSConfigure -action delete realm 59
  AMWLSConfigure -action unconfig 56

W
WebLogic Server
  compatibility mode 9
  Security Service Provider Interface 9
  service packs 9
  Version 7.0 support 9
WebSEAL 1, 10
  authentication 5
  configured user 5
  single sign-on 10, 27
WebSEAL junctions
  configuring 27
Windows
  installing on 16
  removing from 41