Note
Before using this information and the product it supports, read the information in Appendix B, “Notices,” on page 103.
Contents

Preface ............................................................................................................. vii
What should read this book .............................................................................. vii
What this book contains .................................................................................. vii
Publications ...................................................................................................... vii
  Release information ...................................................................................... vii
  Base information ........................................................................................ vii
  Web security information ............................................................................. ix
  Developer references .................................................................................... ix
  Technical supplements ................................................................................ x
  Related publications .................................................................................... x
  Accessing publications online ..................................................................... xiii
Accessibility ...................................................................................................... xiv
Contacting software support .......................................................................... xiv
Conventions used in this book ........................................................................ xiv
  Typeface conventions .................................................................................. xiv
  Operating system differences ..................................................................... xiv

Chapter 1. Introduction and overview ............................................................. 1
  Integrating Tivoli Access Manager with WebSphere Application Server ........ 2
  Java 2 Enterprise Edition role-based security ................................................. 4
  Mapping of principals and groups to roles .................................................... 4
  Centralizing policy management for multiple WebSphere servers .............. 8

Chapter 2. Installation instructions ................................................................. 11
  Software contents ........................................................................................ 11
  Supported platforms ..................................................................................... 11
  Support for WebSphere Application Server version 5.1............................... 12
  Disk and memory requirements ................................................................... 12
  Software prerequisites ................................................................................ 12
    WebSphere Application Server .................................................................. 12
    Tivoli Access Manager Base .................................................................... 13
    Java Runtime Environment ....................................................................... 14
  User registry prerequisites .......................................................................... 14
  Upgrading from a previous release ............................................................... 15
  Installing using the installation wizard ......................................................... 16
  Installing Tivoli Access Manager for WebSphere using native utilities ....... 19
    Installing on Solaris ................................................................................ 19
    Installing on AIX .................................................................................... 20
    Installing on HP-UX ............................................................................. 21
    Installing on Linux ................................................................................ 22
    Installing on Windows. ......................................................................... 23

Chapter 3. Configuration procedures .............................................................. 25
  Configuring the initial installation ................................................................ 25
    Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server .... 26
    Part 2: Enable WebSphere security .......................................................... 27
    Part 3: Configure the Access Manager Java Runtime Environment .......... 29
    Part 4: Join a secure domain ..................................................................... 30
    Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6 . 31
    Part 5b: Migrate WebSphere security settings — WebSphere version 5.0.2 . 34
  Configuring Tivoli Access Manager for WebSphere in a WebSphere Application Server version 5.1 environment .............................................................. 36
    Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server .... 37
    Part 2: Enable Security in WebSphere Application Server version 5.1 ..... 37
    Part 3: Configure the Access Manager Java Runtime Environment .......... 37
    Part 4: Configure Tivoli Access Manager for WebSphere ....................... 37
Part 5: Migrate administration policy ................................................. 38
Configuring additional installations .................................................. 39
Part A-1: Configure the Access Manager Java Runtime Environment ........ 39
Part A-2: Join a secure domain ......................................................... 40

Chapter 4. Migrating security roles .................................................... 43
How to migrate security roles ......................................................... 43
Migration utility limitations ............................................................. 47
Troubleshooting tips ....................................................................... 47
Use of log files ................................................................................. 48
Users not attached to created ACLs .................................................. 48
Migration fails on Windows files with short name .............................. 48
Web Portal Manager unable to attach an ACL to an object ................. 48
Warning that user […] is a member of pdwas-admin ......................... 48
Client authentication lost due to session expiration ......................... 49
Migration utility messages not displayed in correct language ............ 49

Chapter 5. Administration tasks ....................................................... 51
WebSphere Advanced Edition Single Server version 4.0.6 .................... 51
Tivoli Access Manager administration tools ....................................... 52
Specifying runtime properties ......................................................... 52
Configuring static role caching ....................................................... 52
Configure dynamic role caching ....................................................... 53
Role Based Policy Framework Parameters ....................................... 54
Configuring additional authorization servers .................................... 55
Adding an object class to the console ................................................ 56
GSO principal mapping setup .......................................................... 57
Creating a new application login ....................................................... 58
Tivoli Access Manager for WebSphere logging ................................... 61
Single Sign-On to WebSphere Application Server using WebSEAL ....... 63
Step 1 — Create a trusted user account in Tivoli Access Manager ....... 63
Step 2 — Create a WebSEAL junction to the WebSphere Application Server .... 63
Step 3a — Configure SSO using TAI for WebSphere Application Server version 4.0.6 .... 64
Step 3b — Configure SSO using TAI for WebSphere Application Server version 5.0.2 .... 64
Step 4 — Set the SSO password in WebSEAL .................................... 65
Step 5 — Test the WebSEAL connection .......................................... 65
Troubleshooting tips ....................................................................... 65
WebSphere server does not start after configuration and migration — WebSphere Application Server version 4.0.6 only .................. 66
WebSphere server does not start after unconfiguration — WebSphere Application Server version 4.0.6 only .................. 66
Backing up Tivoli Access Manager for WebSphere files .................... 67

Chapter 6. Tutorial: How to enable security ....................................... 69
Tutorial : for Tivoli Access Manager for WebSphere Application Server version 4.0.6 ......................................................... 69
How to use the tutorial .................................................................... 69
Part 1: Add users to the LDAP user registry ....................................... 70
Part 2: Install Tivoli Access Manager for WebSphere ....................... 71
Part 3: Add security to a WebSphere application ............................... 71
Part 4: Create the Tivoli Access Manager administrative user for WebSphere Application Server .......................... 73
Part 5: Enable WebSphere security .................................................. 73
Part 6: Deploy the application ......................................................... 74
Part 7: Test security for the deployed application ............................... 75
Part 8: Migrate the application to Tivoli Access Manager .................. 75
Part 9: Test security for the deployed application ............................... 77
Part 10: Change roles ...................................................................... 77
Part 11: Test security for the deployed application .............................. 78
Tutorial : for Tivoli Access Manager for WebSphere Application Server version 5.0.2 ......................................................... 78
How to use the tutorial .................................................................... 78
Part 1: Add users to the LDAP user registry ....................................... 79
Part 2: Install Tivoli Access Manager for WebSphere ....................... 80
Preface

Welcome to IBM® Tivoli® Access Manager for WebSphere Application Server (Tivoli Access Manager for WebSphere). This product extends Tivoli Access Manager to support applications written for IBM® WebSphere™ Application 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.

The IBM Tivoli Access Manager for WebSphere Application Server Integration Guide provides installation, configuration, and administration instructions. This document also provides a tutorial on configuring centralized security policy for WebSphere applications.

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 WebSphere Application Server systems and applications
- Security management, including authentication and authorization

If you are using 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 Tivoli Access Manager components that provide authorization services to WebSphere Application Server.
- Chapter 2, “Installation instructions”
  Describes how to install Tivoli Access Manager for WebSphere.
- Chapter 3, "Configuration procedures"
Describes how to configure Tivoli Access Manager for WebSphere.

- Chapter 4, “Migrating security roles”
  Describes how to use the Tivoli Access Manager for WebSphere migration utility to migrate Java 2 Enterprise Edition security roles to Tivoli Access Manager user and groups.

- Chapter 5, “Administration tasks”
  Describes how to perform administration tasks that manage Tivoli Access Manager for WebSphere.

- Chapter 6, “Tutorial: How to enable security”
  Describes how to add security to a WebSphere Application Server application. Also describes how to migrate security information to Tivoli Access Manager and how to test that security has been successfully enabled.

- Chapter 7, “Removal instructions”
  Describes how to remove Tivoli Access Manager for WebSphere.

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” on page ix
- “Developer references” on page ix
- “Technical supplements” on page x

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 `padmin` 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/](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/](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 Base CD.
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:


**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

IBM Tivoli Access Manager for WebSphere Application Server (Tivoli Access Manager for WebSphere) is an extension of IBM Tivoli Access Manager (Tivoli Access Manager) that provides container-based authorization and centralized policy management for IBM WebSphere Application Server applications.

Tivoli Access Manager for WebSphere facilitates the use of Tivoli Access Manager providing centralized management of security policy both for WebSphere Application Server resources and for resources that are unrelated to WebSphere Application Server.

Tivoli Access Manager provides management of common identities, user profiles, and authorization mechanisms. Tivoli Access Manager also provides a graphical user interface utility, the Tivoli Access Manager Web Portal Manager, that can be used as a single point of security management both for resources that are compliant with Java™ 2 Enterprise Edition (J2EE) and for resources that are not J2EE compliant.

WebSphere Application Server supports the J2EE security classes and APIs. Tivoli Access Manager for WebSphere supports WebSphere applications that use the J2EE security classes. Tivoli Access Manager for WebSphere provides this support without requiring any coding or deployment changes to the applications.

Tivoli Access Manager for WebSphere can be integrated with WebSphere containers to enable them to use the security services provided by a Tivoli Access Manager secure domain. The secure domain must be deployed prior to installation of Tivoli Access Manager for WebSphere.

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

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, Tivoli Access Manager 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. Note, however, that WebSphere performs its own authentication steps before using Tivoli Access Manager for WebSphere.
- An authorization framework
  The Tivoli Access Manager authorization service, accessed through standard J2EE authorization classes, provides permit and deny decisions on access requests for native Tivoli Access Manager servers and third-party applications.
You can learn more about Tivoli Access Manager, including information necessary to make deployment decisions, by reviewing the product documentation. Start with the following guides:

- **IBM Tivoli Access Manager Base Installation Guide**
  This guide describes how to plan, install, and configure a Tivoli Access Manager secure domain. A series of easy installation scripts enable you to quickly deploy a fully functional secure domain. These scripts are very useful when prototyping the deployment of a secure domain.

- **IBM Tivoli Access Manager Base Administration Guide**
  This document presents an overview of the Tivoli Access Manager security model for managing protected resources. This guide describes how to configure the Tivoli Access Manager servers that make access control decisions. In addition, detailed instructions describe how to perform important tasks such as declaring security policies, defining protected object namespaces, and administering user and group profiles.

## Integrating Tivoli Access Manager with WebSphere Application Server

Tivoli Access Manager for WebSphere extends the Tivoli Access Manager security model to work with applications built for IBM WebSphere Application Server. The security model is used in the following way:

When a user (principal) attempts to access a protected resource, WebSphere performs the following tasks:

- Authenticates the principal.
- When security is specified in the deployment descriptor for an application (declarative security), a WebSphere container determines which roles are required to access the resource, and uses Tivoli Access Manager for WebSphere to determine if the current principal has been granted any of the required roles.
- When an application developer has added security code directly into the application (programmatic security), a WebSphere container uses Tivoli Access Manager to perform the necessary role membership checks.
Figure 1 illustrates the following sequence of events:

1. When a WebSphere application with J2EE security is run, and the user tries to access a protected resource, WebSphere authenticates the user, using the user registry. For example, in Figure 1 WebSphere Advanced Edition, multiple server version, authenticates against an IBM Directory user registry. The user registry is shared with Tivoli Access Manager. (For WebSphere Advanced Edition Single Server, the authentication is against host-based security.)

2. When the user requests access to a protected method or resource, the WebSphere container uses information from the J2EE application deployment descriptor to determine the required role membership.

3. The WebSphere container uses the integrated Tivoli Access Manager module to request an authorization decision (“granted” or “denied”) from the Tivoli Access Manager authorization server.

   The WebSphere container also passes additional context information, when present, to the authorization server. The optional context information can include cell name, host name, and server name. If the Tivoli Access Manager policy database has policies specified for any of the context information, the authorization server can use this information when making the authorization decision.

4. The authorization server consults the Tivoli Access Manager user definitions in the shared user registry. (The user registry is shared with WebSphere, unless WebSphere Advanced Edition Single Server is used). The authorization server then consults the permissions that have been defined for the specified user
within the Tivoli Access Manager protected object namespace. The protected object namespace is included in the policy database shown in Figure 1.

5. The Tivoli Access Manager authorization server returns the access decision to the WebSphere container.

6. WebSphere Application Server either grants or denies access to the protected method or resource.

Java 2 Enterprise Edition role-based security

Java 2 Enterprise Edition (J2EE) security uses the concept of a principal to represent the identity of an entity that performs activities. Entities can be people (users) or processes. In addition, J2EE uses the concept of a role as described below.

Methods are mapped to roles. The following table from a sample banking application defines roles and maps methods to them. The entry granted in the table below indicates that the role can access the specified method.

Table 1. Mapping of methods to roles

<table>
<thead>
<tr>
<th>Roles</th>
<th>Methods</th>
<th>getBalance</th>
<th>deposit</th>
<th>closeAccount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teller</td>
<td>granted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cashier</td>
<td>granted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervisor</td>
<td></td>
<td></td>
<td></td>
<td>granted</td>
</tr>
</tbody>
</table>

The roles that have been defined above can then be mapped to principals, groups, or both. The entry Invoke in the table cells below indicates that the principal or group can invoke any methods that have been granted to that role.

Table 2. Method invocation permissions for principals or groups

<table>
<thead>
<tr>
<th>Principal/Group</th>
<th>Roles</th>
<th>Teller</th>
<th>Cashier</th>
<th>Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TellerGroup</td>
<td>Invoke</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CashierGroup</td>
<td></td>
<td></td>
<td></td>
<td>Invoke</td>
</tr>
<tr>
<td>SupervisorGroup</td>
<td></td>
<td></td>
<td></td>
<td>Invoke</td>
</tr>
<tr>
<td>Frank</td>
<td>Invoke</td>
<td></td>
<td></td>
<td>Invoke</td>
</tr>
</tbody>
</table>

Frank (a principal, is not a member of any of the above groups)

In the table above, the principal Frank can invoke the getBalance and closeAccount methods but cannot invoke the deposit method, because this method has not been granted to either the Cashier or Supervisor role.

Mapping of principals and groups to roles

Prior to application runtime, the Tivoli Access Manager for WebSphere migration utility is run to populate the Tivoli Access Manager protected object namespace. The migration utility obtains information about roles and methods from the J2EE application deployment descriptors.
At application runtime, when a user requests access to a protected resource, the following information is passed to the WebSphere container:

- **Principal**
  The authenticated identity of the user.
- **RoleName**
  The name of a role.
- **AppName**
  The name of the application.
- **CellName**
  The name of a grouping of host systems on the network.
- **HostName**
  The name of a host system contained in CellName.
- **ServerName**
  The name of a server that is hosted by HostName.

The role names are derived from the method-to-role mappings in the deployment descriptors. By default, Tivoli Access Manager’s access check is performed based on the **RoleName** and **AppName**. The access check can easily be extended to take into account **CellName**, **HostName** and **ServerName**. These values are optional, and are evaluated only when they are defined.

Tivoli Access Manager access control lists (ACLs) determine which J2EE application roles have been assigned to a principal. The migration utility attaches ACLs to the **AppName** in the protected object namespace.

Figure 2 below illustrates the following sequence of events:

1. Before application runtime, the Tivoli Access Manager for WebSphere migration utility accesses the J2EE application deployment descriptor to extract information on roles and role-to-principal or role-to-group mapping.
2. The migration utility converts the information into the Tivoli Access Manager format, and passes it to the Tivoli Access Manager policy server.
3. The policy server adds entries to the protected object namespace to represent the roles defined for the application. When role-to-principal or role-to-group mappings have been defined in the deployment descriptor, the appropriate principals or groups are added to the ACLs that are attached to the new objects.
The Tivoli Access Manager security model uses the definitions stored in the protected object namespace to build a hierarchy of resources to which ACLs can be attached. These ACLs define the mapping of roles to users or groups.

Figure 3 below illustrates how ACLs can be applied to the protected object namespace that describes a role. The protected object namespace for all WebSphere applications consists of a top-level protected object called WebAppServer. The WebAppServer object has a child object called deployedResources. Together, these two object names serve as a top-level prefix to all J2EE roles defined in WebSphere applications.
Roles are defined in the next level in the hierarchy, as a resource named for the role: RoleName. Directly under this object is the resource representing the application: AppName. Underneath the AppName protected object are several optional resources that can be defined to more precisely control access to roles. The optional resources are CellName, HostName, and ServerName.

![Tivoli Access Manager Protected Object Namespace](image)

*Figure 3. Attaching ACLs to objects in the protected object namespace.*

In Figure 3 above, **ACL 1** grants user1 access to the specified RoleName, in any application anywhere in the network. User2 and group1 are denied access.

In the Tivoli Access Manager security model, these access settings are inherited by the objects defined underneath RoleName in the protected object space hierarchy. This inheritance occurs by default. Thus, in Figure 3, the access settings are inherited by the objects representing AppName/CellName/HostName/ServerName.

Sometimes security policy requires that the access settings for objects located underneath the ACL attachment point must differ from the inherited access settings. In this case, the Tivoli Access Manager administrator defines a new ACL containing the required access settings. The administrator then attaches the new ACL to the object at the specified point of control. This new ACL overrides the inherited access settings.

For example, security policy might dictate that user1 should not be granted RoleName permission when the application is run on a specific server on a specific host within a specific cell. To enforce this policy, the administrator defines a more restrictive ACL, as represented in Figure 3 by **ACL 2**. This ACL denies access to user1, user2, and grp1. The administrator then attaches this ACL to the ServerName object that represents the server to which access must be restricted.

Figure 3 shows the attachment of **ACL 2** to ServerName. Note that **ACL 2** applies only to the specified server. When more than one ServerName object is defined underneath HostName, **ACL 2** applies only to the ServerName object to which it is attached. All other ServerName objects at this level in the hierarchy continue to inherit the access settings defined in **ACL 1** and attached to RoleName.
For more information on the use of ACLs in the protected object namespace, see the IBM Tivoli Access Manager Base Administration Guide.

Centralizing policy management for multiple WebSphere servers

Tivoli Access Manager provides centralized management of security policies. Tivoli Access Manager can manage security policy across multiple WebSphere Application Servers. In addition, Tivoli Access Manager uses the same model to manage security across non-WebSphere applications.

After the roles and principal or group mappings described in a J2EE application’s deployment descriptors have been migrated to Tivoli Access Manager, and the users and groups have been registered with Tivoli Access Manager, you can use the Tivoli Access Manager management tools to manage further changes to the security definitions. Use the Tivoli Access Manager Web Portal Manager to manage changes in security definitions related to the mapping of roles to principals or groups. Use the WebSphere console to make other security-related changes. Note that changes to role mappings made through the WebSphere console will not be visible to the Tivoli Access Manager security model.

Use the following Tivoli Access Manager tools to manage security policy:

- **Tivoli Access Manager Web Portal Manager**
  The Web Portal Manager is the Tivoli Access Manager management console. This console provides a graphical user interface for managing the Tivoli Access Manager users, actions, and resources that are defined in the Tivoli Access Manager protected object namespace. The console can be used for creating and managing ACLs. The console can also be used to manage user and group definitions in the user registry.

- **pdadmin**
  The pdadmin utility is a command-line based utility for managing the Tivoli Access Manager security model. This powerful utility can be used to manage all aspects of the Tivoli Access Manager protected object namespace, including users, objects, resources, and ACLs. Also, pdadmin can manage user and group entries in user registries. Administrators can use this utility within scripts or programs to automate administration tasks.

  For more information, see the IBM Tivoli Access Manager Base Administration Guide.

- **Tivoli Access Manager Administration API**
  Tivoli Access Manager provides a programmatic interface to the administration tasks accomplished by pdadmin and the Web Portal Manager. Application developers can use a C or Java API to perform administration tasks that are specific to the application.

  For more information, see the IBM Tivoli Access Manager for e-business Administration C API Developer Reference or IBM Tivoli Access Manager for e-business Administration Java Classes Developer Reference.
Figure 4 above illustrates Tivoli Access Manager’s management of security across multiple WebSphere servers. The Web Portal Manager has been installed with WebSphere Application Server on Machine A. The pdadmin utility is shown on a non-WebSphere system, Machine B.

Both the Web Portal Manager and pdadmin use the policy server on Machine D to administer security policy.

The Tivoli Access Manager authorization server can be installed on a system that is separate from the WebSphere system. In Figure 4, Machine E hosts WebSphere Application Server. This server has a Tivoli Access Manager for WebSphere module that has been integrated into the WebSphere container responsible for authorization decisions. The WebSphere container obtains authorization decisions from the Tivoli Access Manager authorization server on Machine F.

The authorization server can also be installed on the same system as the WebSphere Application Server, as shown on Machine G. The Tivoli Access Manager functionality is identical to that provided when the servers are on separate systems (as shown on Machine E and Machine F). Co-location of the authorization server
with the WebSphere Application Server optimizes performance when making authorization decisions. This configuration is recommended.

Note that the Tivoli Access Manager policy database is replicated from Machine D to both Machine F and Machine G. This replication increases performance and provides failover capability.

Figure 4 also shows that the Tivoli Access Manager servers and the WebSphere servers share an LDAP user registry on Machine C. Figure 4 assumes that WebSphere Advanced Edition (multiserver) is being used. The user registry is not shared when using WebSphere Advanced Edition Single Server.
Chapter 2. Installation instructions

This chapter contains the following topics:

- “Software contents”
- “Supported platforms”
- “Disk and memory requirements” on page 12
- “Software prerequisites” on page 12
- “User registry prerequisites” on page 14
- “Upgrading from a previous release” on page 15
- “Installing Tivoli Access Manager for WebSphere using native utilities” on page 19

Software contents

Tivoli Access Manager for WebSphere provides a component that can be integrated with WebSphere Application Server, and takes responsibility for all mappings of roles to principals or groups.

Tivoli Access Manager for WebSphere also provides a migration utility that can be used to import role-to-principal or role-to-group mappings from a Java 2 Enterprise Edition (J2EE) deployment descriptor into an Tivoli Access Manager security schema. This utility can migrate data from either compressed or expanded WebSphere Enterprise Archive (EAR) files.

The Tivoli Access Manager for WebSphere distribution contains the following software:

- Tivoli Access Manager for WebSphere Java classes
- A configuration script called pdwascfg, for the Java classes
- Migration utilities migrateEAR4 and migrateEAR5.
- Sample tutorial code that demonstrates the use of the migration utility and the Java classes

Supported platforms

Tivoli Access Manager for WebSphere is supported on the following platforms for the listed versions of WebSphere Application Server:

- WebSphere Application Server version 4.0.6
  - IBM AIX 5.1 and 5.2
  - Sun Solaris 8
  - HP-UX 11i
  - Microsoft Windows 2000 Server and Advanced Server (Service Pack 3)
  - SuSE SLES8 on IA32
- WebSphere Application Server version 5.0.2
  - IBM AIX 5.1 and 5.2
  - Sun Solaris 8 and 9
  - HP-UX 11i
  - Microsoft Windows 2000 Server and Advanced Servers (Service Pack 3)
Windows 2003 Standard Server and Enterprise Server
SuSE SLES8 on IA32 and zSeries

Support for WebSphere Application Server version 5.1
WebSphere Application Server version 5.1 comes packaged with Tivoli Access Manager for WebSphere. No installation of Tivoli Access Manager for WebSphere is required for customers using WebSphere Application Server version 5.1.

Customers using WebSphere Application Server version 5.1 should ignore the instructions in Chapter 2, “Installation instructions,” on page 11 and following the instructions in “Configuring Tivoli Access Manager for WebSphere in a WebSphere Application Server version 5.1 environment” on page 36.

Disk and memory requirements
Tivoli Access Manager for WebSphere 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 WebSphere Application Server and by any other Tivoli Access Manager components. 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 above and beyond the disk space required by WebSphere Application Server and by any other Tivoli Access Manager components.
- 5 MB disk space for log files.
  This space is in addition to the space required for the software components.

Software prerequisites
The following sections discuss the prerequisites for the integration of Tivoli Access Manager for WebSphere with a WebSphere Application Server environment.

- “WebSphere Application Server”
- “Tivoli Access Manager Base” on page 13
- “Java Runtime Environment” on page 14

WebSphere Application Server
Before you can install Tivoli Access Manager for WebSphere you must have installed one of the following versions of WebSphere Application Server on the host system:

- IBM WebSphere Application Server, Advanced Edition, Version 4.0.6
  or
  IBM WebSphere Application Server, Advanced Edition Single Server, Version 4.0.6
- IBM WebSphere Application Server, Version 5.0.2
- IBM WebSphere Application Server, Version 5.1

The WebSphere Application Server, Advanced Edition, Version 4.0.6 and 5.0.2, and WebSphere Application Server Version 5.1 must be configured to use a user
registry that will be shared with Tivoli Access Manager. The WebSphere users and
groups must be imported into Tivoli Access Manager.

**Note:** The requirement to share a user registry does not apply to WebSphere
Application Server, Advanced Edition Single Server Version 4.0.6. This
version uses host-based security. For more details refer to “WebSphere
Advanced Edition Single Server version 4.0.6” on page 51.

Documentation on installation of the IBM WebSphere Application Server is
available at:
http://www-4.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/nav_pdf.html

If you are new to IBM WebSphere Application Server, consult the *Getting Started
with IBM WebSphere Application Server* guide. This guide is available at the Web site
above.

**Tivoli Access Manager Base**

Tivoli Access Manager for WebSphere requires at least one Tivoli Access Manager
component to be installed on the local host, and requires that a Tivoli Access
Manager secure domain be established. Typically, the secure domain is distributed
across multiple systems.

**Required component on the local host**

Tivoli Access Manager for WebSphere requires that the Access Manager Java
Runtime Environment component be installed on the local computer that hosts the
WebSphere Application Server. This is the minimum Tivoli Access Manager Base
software prerequisite for supporting Tivoli Access Manager for WebSphere.

Tivoli Access Manager for WebSphere does not require any additional Tivoli Access
Manager components on the local computer that hosts the WebSphere Application
Server.

**Optional components on the local host**

While you are not required to add any additional Tivoli Access Manager
components on the local host, you can optimize performance by installing the
Tivoli Access Manager authorization server on the same host as the WebSphere
Application Server. The Tivoli Access Manager runtime environment is a
prerequisite for the authorization server. If you install the authorization server on
the WebSphere host then the Tivoli Access Manager runtime environment must
also be installed on this machine. Both of these components are distributed as part
of the Tivoli Access Manager Base product.

**Tivoli Access Manager secure domain**

Tivoli Access Manager for WebSphere must be able to access a Tivoli Access
Manager secure domain. The migration tool must be able to contact a Tivoli Access
Manager policy server. For best performance, it is recommended that one or more
Tivoli Access Manager authorization servers also be installed in the secure domain.
Thus, after you have installed the IBM WebSphere Application Server, you must
establish a secure domain before installing Tivoli Access Manager for WebSphere.

To establish a secure domain, you must install and configure the policy server.
Typically this is not run on the same host as the WebSphere Application Server.
You can install and configure an authorization server either on the WebSphere
Application Server host or on a different system.
For more information on installing and configuring a Tivoli Access Manager secure domain, including the Access Manager Java Runtime Environment, see the IBM Tivoli Access Manager Base Installation Guide.

**Java Runtime Environment**

The computer system that hosts Tivoli Access Manager for WebSphere must have the Java Runtime Environment version 1.3.1 installed.

The Java Runtime Environment is installed and configured as part of the IBM WebSphere Application Server installation. Tivoli Access Manager for WebSphere uses the same Java runtime environment.

**Note:** Tivoli Access Manager for WebSphere also makes use of the Access Manager Java Runtime Environment. The Access Manager Java Runtime Environment extends the Version 1.3.1 Java runtime.

---

**User registry prerequisites**

Tivoli Access Manager for WebSphere operates as part of a Tivoli Access Manager secure domain. The policy server for the secure domain uses a user registry to manage user and group information.

Tivoli Access Manager for WebSphere supports all of the user registry types that are supported by Tivoli Access Manager Base:

- IBM Directory Server
- Sun ONE Directory Server
- IBM Lotus Domino Server
- Microsoft Active Directory including the non-application version.
- Novell eDirectory

For a complete list of supported versions of each user registry type, see the IBM Tivoli Access Manager Base Installation Guide.

Sun ONE Directory Server version 5.1 and 5.2 is supported by Tivoli Access Manager Base but not by WebSphere Application Server 4.0.6 or 5.0.2. See the software prerequisites pages for WebSphere Application Server for confirmation of versions:

WebSphere Application Server version 4.0.6

WebSphere Application Server version 5.0.2
[http://www.ibm.com/software/webservers/appserv/doc/v50/prereqs/was_v502.htm](http://www.ibm.com/software/webservers/appserv/doc/v50/prereqs/was_v502.htm)

The user registry prerequisites for each installation are also based on the version of WebSphere Application Server that is used with Tivoli Access Manager for WebSphere.


There are two prerequisites for use of user registries that must be satisfied before installing Tivoli Access Manager for WebSphere:
- The Tivoli Access Manager policy server and the WebSphere Application Server must be configured to use the same user registry.
- Any existing users and groups defined for WebSphere Application Server must be imported into the Tivoli Access Manager user directory, to become Tivoli Access Manager users and groups. Importing here means adding extended Tivoli Access Manager attributes, along with the existing user and group definitions, into the Tivoli Access Manager security schema.

Users can be imported into the Tivoli Access Manager user registry manually using the pdadmin command. Tivoli Access Manager secure domains that use IBM Directory LDAP can make use of the Directory bulk load feature.

For more information on using the pdadmin command to import users manually, see the IBM Tivoli Access Manager Base Administration Guide.

For more information on the bulk loading of IBM Directory users, see the IBM Tivoli Access Manager for e-business Performance Tuning Guide.

- WebSphere Application Server, Advanced Edition Single Server, Version 4.0.6

WebSphere Advanced Edition Single Server does not use any external user registry. Instead, it works with host-based security. Each user account on the host system must have an equivalent entry in the user registry used by Tivoli Access Manager.

**Note:** Any changes made over time to the host-based security must also be made to the user registry used by Tivoli Access Manager.

---

**Upgrading from a previous release**

Tivoli Access Manager for WebSphere can be upgraded from the following previous releases:

- IBM Tivoli Access Manager for WebSphere Application Server, Version 3.9
- IBM Tivoli Access Manager for WebSphere Application Server, Version 4.1

The upgrade process consists of unconfiguring the previous release, removing the previous release, and then installing Tivoli Access Manager for WebSphere Version 5.1 and configuring it.

To upgrade Tivoli Access Manager for WebSphere, complete the following steps:

1. Unconfigure and remove the previous release. Follow the removal instructions for your operating system in the appropriate user guide:

2. Upgrade the prerequisite Tivoli Access Manager base packages and secure domain from Version 3.9 or Version 4.1 to Version 5.1.

Determine which Tivoli Access Manager base packages are installed on the computer that hosts Tivoli Access Manager for WebSphere. Each deployment includes at a minimum the Access Manager Java Runtime Environment.

Depending on the topology of the Tivoli Access Manager secure domain, the host computer might also include:

- Tivoli Access Manager runtime environment
- Tivoli Access Manager policy server
- Tivoli Access Manager authorization server
When the local computer system does not include the policy server or authorization server, you must first upgrade the secure domain on the computer system that hosts those servers. When the policy server and authorization server are upgraded to Version 5.1, you can then upgrade the Access Manager Java Runtime Environment package on the local computer.

When the local computer contains the policy server and authorization server, you can upgrade all Tivoli Access Manager base packages at one time.

For instructions on upgrading Tivoli Access Manager base packages and the secure domain, see IBM Tivoli Access Manager Base Installation Guide. Complete the instructions in that document before continuing to the next step.

3. Install the current version of Tivoli Access Manager for WebSphere. Follow the steps in “Installing using the installation wizard” or “Installing Tivoli Access Manager for WebSphere using native utilities” on page 19. After installation, configure the software.

---

**Installing using the installation wizard**

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

- Access Manager Java Runtime Environment
- Tivoli Access Manager for WebSphere Application Server

Before executing the installation Wizard you will need to set the WAS_HOME environment variable on UNIX and Windows, and the PDWAS_HOME environment variable on UNIX.

To set the WAS_HOME environment variable to the WebSphere Application Server installation directory, change directory to WebSphere_install_directory/bin and run the following command:

**UNIX**

```bash
setupCmdLine.sh
```

**Windows**

```bash
setupCmdLine.bat
```

On UNIX platforms, set the PDWAS_HOME environment variable to the location where Tivoli Access Manager for WebSphere is to be installed:

```bash
PDWAS_HOME=/opt/amwas
export PDWAS_HOME
```

Ensure that the java /bin directory in the Java Runtime shipped with WebSphere Application Server is the first in the system path.

To install and configure a Tivoli Access Manager for WebSphere Application Server using the install_amwas 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 “Software prerequisites” on page 12.
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 WebSphere Application Server is installed and configured on this machine.

5. Perform the configuration steps as outlined in:
   - “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26
   - “Part 2: Enable WebSphere security” on page 27
   - “Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6” on page 31 or “Part 5b: Migrate WebSphere security settings — WebSphere version 5.0.2” on page 34 depending on the version of WebSphere Application Server you are running.

6. On Windows systems, exit from all running programs before initiating the installation wizard.

7. Stop WebSphere Application Server.

8. Run the install_amwas program, located in the root directory on Tivoli Access Manager Web Security CD for AIX, HP-UX, Linux, Solaris, and Windows platforms.

   **Note:** If WebSphere application server is not installed in the default location, specify its location when running the install_amwas command by using the -is javahome option. For example:

   ```
   install_amwas -is:javahome websphere_install_dir/AppServer/java/jre
   ```

   The installation Wizard begins.

   a. The **Choose Setup Language** dialog is displayed. Select the appropriate language and click **OK**.

   b. The **Welcome** dialog is displayed. Click **Next**.

   c. At the **License Agreement** dialog, click **I Agree** if you accept the license terms and conditions.

   d. If you have not installed the Tivoli Access Manager Base components you are prompted to install them now. Click **Next** to continue with the AMJRTE Installation. If AMJRTE is already installed, the installation Wizard continues at step e. Otherwise, the **AMJRTE Environment Configuration Window** is displayed. Enter the configuration values using the following table as a guide.

<table>
<thead>
<tr>
<th>Configuration Options</th>
<th>Description</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy server host name*</td>
<td>The fully qualified host name of the policy server.</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>For example: pdmgr.tivoli.com</td>
<td></td>
</tr>
<tr>
<td>Policy server SSL port*</td>
<td>The port number on which the policy server listens for SSL requests.</td>
<td>7135</td>
</tr>
<tr>
<td>JRE directory*</td>
<td>The path of the JRE that was installed and shipped with WebSphere Application Server. If you installed using the -is:javahome option the path shown is that specified as the javahome option.</td>
<td>n/a</td>
</tr>
</tbody>
</table>

   * indicates a required option

   e. You are prompted to enter the Tivoli Access Manager for WebSphere Application Server installation directory. Accept the default value and click **Next**.

   f. You are prompted to enter configuration items. Enter the values using the following table as a guide.
Table 3. **install_amwas** installation wizard configuration options.

<table>
<thead>
<tr>
<th>Configuration Options</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remote ACL User</strong> *</td>
<td>The name used to create the Access Manager Application ID used by Tivoli Access Manager for WebSphere to perform authorization checking. For example: pdperadmin</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>sec_master password</strong> *</td>
<td>The password for the Tivoli Access Manager sec_master administrator account.</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Policy server host name</strong> *</td>
<td>The fully qualified host name of the policy server. For example: pdmgr.tivoli.com</td>
<td></td>
</tr>
<tr>
<td><strong>Policy server port number</strong> *</td>
<td>The port number on which the policy server listens for requests.</td>
<td>7135</td>
</tr>
<tr>
<td><strong>Authorization server host name</strong> *</td>
<td>The host name of the authorization server to be used by Tivoli Access Manager for WebSphere. It is recommended that this host name be the same as the WebSphere host name. For example: pdacld.tivoli.com</td>
<td></td>
</tr>
<tr>
<td><strong>Authorization server port number</strong> *</td>
<td>The port number on which the authorization server listens for SSL requests.</td>
<td>7136</td>
</tr>
<tr>
<td><strong>Type of configuration to perform.</strong> Set to true if this product was packaged with WebSphere.</td>
<td>all, local or remote. This should always be set to false when installing using the installation Wizard.</td>
<td>all</td>
</tr>
<tr>
<td><strong>The version of WebSphere Application Server installed</strong> *</td>
<td>The version of WebSphere Application Server that is installed. The choices are: WAS5 or WAS4.</td>
<td>WAS5</td>
</tr>
<tr>
<td><strong>The Tivoli Access Manager for WebSphere Application Server installation directory.</strong></td>
<td>The directory where you want to install Tivoli Access Manager for WebSphere. This defaults to the value entered in the previous installation window.</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>The WebSphere Application Server install directory</strong> *</td>
<td>The directory where WebSphere Application Server is installed. This should be set to the same value as the WAS_HOME environment variable.</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>The URL to the JRTE properties file that will be configured.</strong></td>
<td>The URL path to the AMJRTE PdPerm.properties</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>The URL of the AMJRTE keystore file.</strong></td>
<td>The URL path to the AMJRTE keystore used internally for communication with the policy and authorization servers.</td>
<td>n/a</td>
</tr>
</tbody>
</table>

* indicates a required option
Note: For Windows installations, when prompted for configuration options for the Access Manager Runtime Environment, ensure that you specify the JRE directory for the JRE shipped and installed with WebSphere Application Server. For example:

\texttt{websphere_install_dir\AppServer\java\jre}

### Installing Tivoli Access Manager for WebSphere using native utilities

This section describes how to install Tivoli Access Manager for WebSphere, including both the authorization component and the migration utility.

Complete the instructions that apply to your operating system:

- “Installing on Solaris”
- “Installing on AIX” on page 20
- “Installing on HP-UX” on page 21
- “Installing on Linux” on page 22
- “Installing on Windows” on page 23

### Installing on Solaris

The Tivoli Access Manager for WebSphere installation separates file extraction from package configuration. Use \texttt{pkgadd} to install software packages on Solaris.

Note: If you have already installed and configured Tivoli Access Manager for WebSphere and need to reinstall it, you must first unconfigure and remove it. See “Removing from Solaris” on page 87.

To install Tivoli Access Manager for WebSphere on Solaris complete the following instructions:

1. Log in as user \texttt{root}.
2. Verify that you have satisfied the prerequisites for installing Tivoli Access Manager for WebSphere.

   To review software dependencies, see “Software prerequisites” on page 12.

3. Verify that the Tivoli Access Manager policy server and the WebSphere Application Server are configured to use the same user registry.

   Note: This step does not apply to WebSphere Advanced Edition Single Server.

   To review user registry dependencies, see “User registry prerequisites” on page 14.

4. Verify that WebSphere Application Server users and groups have been imported from the user registry into the Tivoli Access Manager user registry schema.

   You can use the Tivoli Access Manager \texttt{pdadmin} command to manually import users. For example, the syntax for importing an LDAP user is:

   \texttt{pdadmin> user import UserID Distinguished_Name_of_the_user_in_LDAP}

   For more information on \texttt{pdadmin}, see the IBM Tivoli Access Manager Base Administration Guide.

   For large numbers of users in an IBM Directory LDAP environment, consider using the LDAP bulk import feature. For more information see the IBM Tivoli Access Manager for e-business Performance Tuning Guide.
5. Insert the IBM Tivoli Access Manager Web Security for Solaris CD.
6. Install the following packages (one at a time):
   pkgadd -d /cdrom/cdrom0/solaris -a /cdrom/cdrom0/solaris/pddefault packages
   
   where:
   • -d /cdrom/cdrom0/solaris — Specifies the location of the package.
   • -a /cdrom/cdrom0/solaris/pddefault — Specifies the location of the installation administration script.
   
   and packages are as follows:
   • PDJrte — The Access Manager Java Runtime Environment package.
   • PDWAS — The Tivoli Access Manager WebSphere Application Server package.
   
   Note: These packages must be installed on the same system as WebSphere Application Server.

Installing on AIX

The Tivoli Access Manager for WebSphere installation separates file extraction from package configuration.

Note: If you have already installed and configured Tivoli Access Manager for WebSphere and need to reinstall it, you must first unconfigure and remove the Tivoli Access Manager for WebSphere package. See “Removing from AIX” on page 88.

To install Tivoli Access Manager for WebSphere on AIX complete the following instructions:
1. Log in as root.
2. Verify that you have satisfied the prerequisites for installing Tivoli Access Manager for WebSphere.
   
   To review software dependencies, see “Software prerequisites” on page 12.
3. Verify that the Tivoli Access Manager policy server and the WebSphere Application Server are configured to use the same user registry.
   
   Note: This step does not apply to WebSphere Advanced Edition Single Server.
   
   To review user registry dependencies, see “User registry prerequisites” on page 14.
4. Verify that WebSphere Application Server users and groups have been imported from the user registry into the Tivoli Access Manager user registry schema.
   
   You can use the Tivoli Access Manager pdadmin command to manually import users. For example, the syntax for importing an LDAP user is:
   pdadmin> user import UserID Distinguished_Name_of_the_user_in_LDAP

   For more information on pdadmin, see the IBM Tivoli Access Manager Base Administration Guide.
For large numbers of users in an IBM Directory LDAP environment, consider using the LDAP bulk import feature. For more information see the IBM Tivoli Access Manager for e-business Performance Tuning Guide.

5. Insert the IBM Tivoli Access Manager Web Security for AIX CD into the CD drive.

6. Install the following packages:
   
   ```bash
   installp -acgXd cd_mount_point/usr/sys/inst.images packages
   ```
   
   where `cd_mount_point/usr/sys/inst.images` is the directory where the CD is mounted and packages are as follows:

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDJ.rte</td>
<td>Specifies the Access Manager Java Runtime Environment package.</td>
</tr>
<tr>
<td>PDWAS</td>
<td>Specifies the Access Manager for WebLogic Application Server package.</td>
</tr>
</tbody>
</table>

   **Note:** These packages must be installed on the same system as WebSphere Application Server.


**Installing on HP-UX**

The Tivoli Access Manager for WebSphere installation separates file extraction from package configuration. Use `swinstall` to install software packages on HP-UX.

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

1. Log in as user `root`.

2. Verify that you have satisfied the prerequisites for installing Tivoli Access Manager for WebSphere.

   To review software dependencies, see “Software prerequisites” on page 12.

3. Verify that the Tivoli Access Manager policy server and the WebSphere Application Server are configured to use the same user registry.

   **Note:** This step does not apply to WebSphere Advanced Edition Single Server.

4. Verify that WebSphere Application Server users and groups have been imported from the user registry into the Tivoli Access Manager user registry schema.

   You can use the Tivoli Access Manager `padmin` command to manually import users. For example, the syntax for importing an LDAP user is:

   ```bash
   padmin> user import UserID Distinguished_Name_of_the_user_in_LDAP
   ```

   For more information on `padmin`, see the IBM Tivoli Access Manager Base Administration Guide.

   For large numbers of users in an IBM Directory LDAP environment, consider using the LDAP bulk import feature. For more information see the IBM Tivoli Access Manager for e-business Performance Tuning Guide.

5. 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:
6. Enter the following command to install the Tivoli Access Manager for WebSphere package:

```
# swinstall -s /cd-rom/hp packages
```

where packages are as follows:

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDJ.rte</td>
<td>Specifies the Access Manager Java Runtime Environment package.</td>
</tr>
<tr>
<td>PDWAS</td>
<td>Specifies the Access Manager for WebSphere Application Server package.</td>
</tr>
</tbody>
</table>

Note: These packages must be installed on the same system as WebSphere Application server.

A message indicates that the analysis phase has succeeded. Another message indicates that the execution phase is beginning. Files are extracted from the CD and installed on the hard disk. A message indicates that the execution phase has succeeded. The `swinstall` utility exits.


## Installing on Linux

The Tivoli Access Manager for WebSphere installation separates file extraction from package configuration. Use `rpm` to install software packages on Linux.

If you have already installed and configured Tivoli Access Manager for WebSphere and need to reinstall it, you must first unconfigure and remove it. See “Removing from Linux” on page 89.

Note: Linux on zSeries users: You must first obtain access to the Linux `rpm` files from a IBM Tivoli Access Manager for Linux on zSeries CD.

To install Tivoli Access Manager for WebSphere on Linux complete the following instructions:

1. Log in as user `root`.
2. Verify that you have satisfied the prerequisites for installing Tivoli Access Manager for WebSphere.
   
   To review software dependencies, see “Software prerequisites” on page 12.
3. Verify that the Tivoli Access Manager policy server and the WebSphere Application Server are configured to use the same user registry.

   Note: This step does not apply to WebSphere Advanced Edition Single Server.

   To review user registry dependencies, see “User registry prerequisites” on page 14.
4. Verify that WebSphere Application Server users and groups have been imported from the user registry into the Tivoli Access Manager user registry schema.

   You can use the Tivoli Access Manager `pdadmin` command to manually import users. For example, the syntax for importing an LDAP user is:
pdadmin> user import UserID Distinguished_Name_of_the_user_in_LDAP

For more information on **pdadmin**, see the *IBM Tivoli Access Manager Base Administration Guide*.

For large numbers of users in an IBM Directory LDAP environment, consider using the LDAP bulk import feature. For more information see the *IBM Tivoli Access Manager for e-business Performance Tuning Guide*.

5. Mount the *IBM Tivoli Access Manager Web Security CD* for xSeries or zSeries.

6. Change to the `/mnt/cdrom/series` directory where `/mnt/cdrom` is the mount point for your CD and `series` specifies xSeries, zSeries, iSeries, or pSeries.

7. Install the following packages:

   ```bash
   rpm -ihv packages
   ```

   where `packages` specifies one of the following:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Access Manager Java Runtime Environment package</th>
<th>Access Manager for WebSphere Application Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux on xSeries</td>
<td>PDJrte-PD-5.1.0-0.i386.rpm</td>
<td>PDWAS-PD-5.1.0-0.i386.rpm</td>
</tr>
<tr>
<td>Linux on zSeries</td>
<td>PDJrte-PD-5.1.0-0.i390.rpm</td>
<td>PDWAS-PD-5.1.0-0.i390.rpm</td>
</tr>
</tbody>
</table>

   **Note:** These packages must be installed on the same system as WebSphere Application Server.


### Installing on Windows

The Tivoli Access Manager for WebSphere installation separates file extraction from package configuration. Use an InstallShield `setup.exe` to install the Tivoli Access Manager for WebSphere files.

**Note:** If you have already installed and configured Tivoli Access Manager for WebSphere and need to reinstall it, you must first unconfigure and remove it. See “Removing from Windows” on page 88.

To install and configure Tivoli Access Manager for WebSphere on Windows complete the following instructions:

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

2. Verify that you have satisfied the prerequisites for installing Tivoli Access Manager for WebSphere.

   To review software dependencies, see “Software prerequisites” on page 12.

3. Verify that the Tivoli Access Manager policy server and the WebSphere Application Server are configured to use the same user registry.

   **Note:** This step does not apply to WebSphere Advanced Edition Single Server.

   To review user registry dependencies, see “User registry prerequisites” on page 14.

4. Verify that WebSphere Application Server users and groups have been imported from the user registry into the Tivoli Access Manager user registry schema.
You can use the Tivoli Access Manager pdadmin command to manually import users. For example, the syntax for importing an LDAP user is:

```
pdadmin> user import UserID Distinguished_Name_of_the_user_in_LDAP
```

For more information on pdadmin, see the IBM Tivoli Access Manager Base Administration Guide.

For large numbers of users in an IBM Directory LDAP environment, consider using the LDAP bulk import feature. For more information see the IBM Tivoli Access Manager for e-business Performance Tuning Guide.

5. Insert the IBM Tivoli Access Manager Web Security for Windows CD into the CD drive.
6. Install the Access Manager Java Runtime Environment and Access Manager for WebSphere Application Server packages. To do so, run the setup.exe file in the following directory:
   \windows\PolicyDirector\Disk Images\Disk1\setup.exe

   The Choose Setup Language dialog is displayed.
7. Select the language that you want to use for the installation and click OK.
8. The Welcome dialog is displayed. Click Next to continue.
9. Read the license agreement and click Yes if you agree to the terms.
10. Select the following packages and click Next:
    • Access Manager Java Runtime Environment
    • Access Manager for WebSphere Application Server
11. Accept the default destination directory or click Browse to select a path to another directory on the local system. If the directory does not exist, you must confirm that you want the directory created or specify a directory that exists.
12. Click Finish to exit the setup program.
Chapter 3. Configuration procedures

The configuration steps for Tivoli Access Manager for WebSphere differ depending on whether you are configuring the first Tivoli Access Manager for WebSphere system into a Tivoli Access Manager secure domain or you are adding an additional Tivoli Access Manager for WebSphere system.

Each Tivoli Access Manager for WebSphere system is configured into the secure domain by using the pdwascfg utility. Security information for J2EE applications must be migrated to the Tivoli Access Manager policy database. Tivoli Access Manager for WebSphere provides a migration utility to do this. Note that this needs to be performed only on systems that have J2EE applications with EAR files that specify the security policy.

In addition, there are several configuration steps that are required only when configuring the first Tivoli Access Manager for WebSphere system into a given Tivoli Access Manager secure domain.

Continue to one of the following sections:
• “Configuring the initial installation”
• “Configuring Tivoli Access Manager for WebSphere in a WebSphere Application Server version 5.1 environment” on page 36
• “Configuring additional installations” on page 39

Configuring the initial installation

This section describes how to configure the first Tivoli Access Manager for WebSphere installation.

Tivoli Access Manager for WebSphere provides utilities to quicken the configuration process. The configuration steps use these utilities, plus the Tivoli Access Manager administrative utility pdadmin and the WebSphere console. A number of the steps need be performed only the first time Tivoli Access Manager for WebSphere is configured into a specific Tivoli Access Manager secure domain.

The configuration instructions are described in the following sections:
• “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26
• “Part 2: Enable WebSphere security” on page 27
• “Part 3: Configure the Access Manager Java Runtime Environment” on page 29
• “Part 4: Join a secure domain” on page 30
• “Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6” on page 31
• “Part 5b: Migrate WebSphere security settings — WebSphere version 5.0.2” on page 34

The configuration steps for initial configuration into a secure domain are summarized in the diagram below.
Complete the instructions in each of the numbered parts in this section.

**Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server**

If security has already been enabled in WebSphere Application Server the WebSphere Application Server administrative user should be imported into the Tivoli Access Manager object space. Use either the Tivoli Access Manager command line utility, `pdadmin`, or the Tivoli Access Manager Web Portal Manager to import the Tivoli Access Manager administrative user for WebSphere Application Server. To do this from the Tivoli Access Manager command line utility:

1. From a command line, start `pdadmin` as administrative user `sec_master`:
   ```bash
   pdadmin -a sec_master -p sec_master_password
   ```
2. Import the WebSphere Application Server administrative user. For example:
   ```bash
   pdadmin> user import was_admin_user dn_registry_identifier
   ```

   Make the WebSphere administrative user account valid:
   ```bash
   pdadmin> user modify was_admin_user account-valid yes
   ```

If security has not been enabled in WebSphere Application Server the WebSphere Application Server administrative user needs to be created. Use either the Tivoli Access Manager command line utility, `pdadmin`, or the Tivoli Access Manager Web Portal Manager to create the Tivoli Access Manager administrative user for WebSphere Application Server.

The following instructions describe how to use `pdadmin`.

1. From a command line, start `pdadmin` as administrative user `sec_master`:
   ```bash
   pdadmin -a sec_master -p sec_master_password
   ```
2. Create a Tivoli Access Manager administrative user for WebSphere Application Server. For example, the following instructions create a new user `wsadmin`. The following command must be entered as one continuous command line:
   ```bash
   pdadmin> user create wsadmin cn=wsadmin,o=organization,c=country
   wsadmin wsadmin myPassword
   ```

---

**Figure 5. Configuration tasks for initial installation of Tivoli Access Manager for WebSphere.**

<table>
<thead>
<tr>
<th>TASK</th>
<th>UTILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 1: Create an administrative user</td>
<td><code>pdadmin</code></td>
</tr>
<tr>
<td>Part 2: Enable WebSphere security</td>
<td><code>WebSphere console</code></td>
</tr>
<tr>
<td>Part 3: Configure the Tivoli Access Manager Java runtime</td>
<td><code>pdjrtecfg</code></td>
</tr>
<tr>
<td>Part 4: Join a secure domain</td>
<td><code>pdwascfg</code></td>
</tr>
<tr>
<td>Part 5a or 5b: Migrate WebSphere security settings</td>
<td><code>migrateEAR</code></td>
</tr>
</tbody>
</table>

---

26 IBM Tivoli Access Manager for e-business: IBM WebSphere Application Server Integration Guide
Substitute values for organization and country that are valid for your LDAP user registry.

Make the wsadmin account valid:
```
padmin> user modify wsadmin account-valid yes
```

**Part 2: Enable WebSphere security**

Complete the steps in one of the following sections depending on the version of WebSphere Application Server you are using:

- “Enable security in WebSphere Application Server version 4.0.6”
- “Enable security in WebSphere Application Server version 5.0.2”

Instructions for enabling WebSphere version 5.1 security are included as part of "Configuring Tivoli Access Manager for WebSphere in a WebSphere Application Server version 5.1 environment" on page 36.

**Enable security in WebSphere Application Server version 4.0.6**

To enable security in WebSphere Application Server version 4.0.6:

1. Start the WebSphere Administration Server.
2. When the server has started, start the WebSphere Administration Client.
3. Select **Console > Security Center**.
4. Select the **General** tab. Check the **Enable Security** box.
5. Select the **Authentication** tab.
   a. Select **LTPA**. Set the following LTPA settings:
      - **Token Expiry**: 120
      - **Domain**: *Your domain name*. For example: mydomain.ibm.com
   b. Check the **LDAP** check box. Assign the LDAP settings:

<table>
<thead>
<tr>
<th>LDAP Settings</th>
<th>Example Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Server ID</td>
<td>cn=wsadmin,o=ibm,c=us</td>
</tr>
<tr>
<td>Security Server Password</td>
<td>myPassword</td>
</tr>
<tr>
<td>Host</td>
<td>ldapserver.mydomain.ibm.com</td>
</tr>
<tr>
<td>Directory Type</td>
<td>SecureWay</td>
</tr>
<tr>
<td>Base DN</td>
<td>o=ibm,c=us</td>
</tr>
<tr>
<td>Bind DN</td>
<td>cn=root</td>
</tr>
<tr>
<td>Bind Password</td>
<td>myPassword</td>
</tr>
</tbody>
</table>

c. Click OK.
6. Right click on **WebSphere Admin Domain > Nodes > Hostname**
7. Select **Restart**.

**Enable security in WebSphere Application Server version 5.0.2**

To enable security in WebSphere Application Server version 5.0.2:

1. Start the WebSphere Administration Server:
2. When the server has started, open the Administrative Console — http://localhost:9090/admin/
3. Login as any user.
4. Configure LDAP:
   a. Select Security → User Registries → LDAP
   b. Configure the following values:

<table>
<thead>
<tr>
<th>LDAP Settings</th>
<th>Example Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server User ID</td>
<td>cn=wsadmin,o=ibm,c=us</td>
</tr>
<tr>
<td>Server User Password</td>
<td>myPassword</td>
</tr>
<tr>
<td>Type</td>
<td>IBM_Directory_Server</td>
</tr>
<tr>
<td>Host</td>
<td>ldapservg_server.mydomain.ibm.com</td>
</tr>
<tr>
<td>Port</td>
<td>389</td>
</tr>
<tr>
<td>Base DN</td>
<td>o=ibm,c=us</td>
</tr>
<tr>
<td>Bind DN</td>
<td>cn=root</td>
</tr>
<tr>
<td>Bind Password</td>
<td>myPassword</td>
</tr>
<tr>
<td>Search Timeout</td>
<td>120</td>
</tr>
<tr>
<td>Reuse connection</td>
<td>true</td>
</tr>
<tr>
<td>Ignore case</td>
<td>true</td>
</tr>
<tr>
<td>SSL Enabled</td>
<td>false</td>
</tr>
<tr>
<td>SSL Configuration</td>
<td>cellname/DefaultSSLSettings</td>
</tr>
</tbody>
</table>

   c. Click Apply.

5. Configure LTPA authentication:
   a. Select Authentication Mechanisms → LTPA
   b. Set the password to encrypt and decrypt LTPA keys.
   c. Set the LTPA expiration key timeout value as 120.
   d. In the same window, confirm password to encrypt and decrypt LTPA keys.
   e. Click Apply.
   f. From the Additional Properties section at the bottom of the screen, select Single Signon (SSO).
   g. Enable single signon.
   h. Enter the single signon DNS domain name.
   i. Click Apply.

6. Configure security settings:
   a. Select Security → Global Security
   b. Configure the following values:

<table>
<thead>
<tr>
<th>Security setting</th>
<th>Example Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>true</td>
</tr>
<tr>
<td>Enforce Java 2 Security</td>
<td>false</td>
</tr>
<tr>
<td>Use domain qualified user IDs</td>
<td>true</td>
</tr>
<tr>
<td>Cache timeout</td>
<td>600</td>
</tr>
<tr>
<td>Issue permission warning</td>
<td>true</td>
</tr>
<tr>
<td>Active protocol</td>
<td>CSI and SAS</td>
</tr>
<tr>
<td>Active authentication mechanism</td>
<td>LTPA</td>
</tr>
</tbody>
</table>
Table 6. Security settings (continued)

<table>
<thead>
<tr>
<th>Active user registry</th>
<th>LDAP</th>
</tr>
</thead>
</table>

- Click **Apply**.

7. Click on the **Save** link.
8. Click on the **Save** button to save the Master Configuration.
9. Log out of the WebSphere Application Server Administration Console.
10. Restart the WebSphere Application Server.

**Part 3: Configure the Access Manager Java Runtime Environment**

Configure the Access Manager Java Runtime Environment to extend the Java runtime that is distributed with IBM WebSphere Application Server.

**Note:** The Access Manager Java Runtime Environment is a software prerequisite for Tivoli Access Manager for WebSphere.

The Access Manager Java Runtime Environment can be configured using either the Access Manager Base configuration GUI or from the command line using the `pdjrtecfg` command. To configure the Access Manager Java Runtime Environment from the Access Manager Base configuration GUI:

1. Change directory to the following location:
   - (UNIX) /opt/PolicyDirector/bin
   - (Windows) C:\Program Files\Tivoli\Policy Director\bin
2. Enter the following command:
   `pdconfig`
   The **Access Manager Configuration** screen is displayed from which you can configure the Java runtime.

To configure the Access Manager Java Runtime Environment from the command line:

1. Verify that environment variable `WAS_HOME` is set to the IBM WebSphere Application Server home directory.
2. Change directory to the following location:
   - (UNIX) /opt/PolicyDirector/sbin
   - (Windows) C:\Program Files\Tivoli\Policy Director\sbin
3. Enter the following command as one continuous line:
   - **UNIX**
     ```
     pdjrtecfg -action config
     -java_home $WAS_HOME/java/jre
     -host policy_server_host
     ```
   - **Windows**
     ```
     pdjrtecfg -action config
     -java_home %WAS_HOME%\java\jre
     -host policy_server_host
     ```

   **Note:** Ensure that the location of the `java` binary that appears first in your `PATH` variable matches the location of the `java` binary that you specify to the `pdjrtecfg` option `-java_home pathname`. 

Chapter 3. Configuration procedures 29
Part 4: Join a secure domain

Complete the following steps:

1. Stop WebSphere Application Server.
2. Assemble the following information:
   - The name of the user account you want to use as a user identity for the Tivoli Access Manager for WebSphere application. The example commands in these instructions use the identity pdpermadmin. The user name you choose should not exist in the user registry.
   - The password for the sec_master account.
   - The fully qualified domain name for the computer that hosts the policy server. For example: pdmgrserver.mysubnet.ibm.com
   - The fully qualified domain name for the computer that hosts the authorization server. For example: pdacldserver.mysubnet.ibm.com
   - The home directory of the WebSphere installation.
3. To set the WAS_HOME environment variable to the WebSphere Application Server installation directory, change directory to WebSphere_install_directory/bin and run the following command:
   
   **UNIX**
   ```bash
   setupCmdLine.sh
   ```
   
   **Windows**
   ```bash
   setupCmdLine.bat
   ```
4. On UNIX platforms, set the PDWAS_HOME environment variable to the Tivoli Access Manager for WebSphere installation directory. On Windows platforms, PDWAS_HOME will already exist in the environment.
   
   **UNIX**
   ```bash
   PDWAS_HOME=/opt/amwas
   export PDWAS_HOME
   ```
5. Change directory to:
   - (UNIX) /opt/amwas/sbin
   - (Windows) C:\Program Files\Tivoli\amwas/sbin
6. Run the pdwascfg utility. Use the information you collected in the previous step to supply the command line options to `pdwascfg`.

   **Note:** The example commands below assume that you are creating a new Tivoli Access Manager user account called pdpermadmin. For example:
   ```bash
   -remote_acl_user pdpermadmin
   ```

   Using the parameters assembled previously, enter the following command, as one continuous command line, using either the -action configWAS4 or configWAS5 parameter depending on the version of WebSphere Application Server that you are using:
   ```bash
   pdwascfg -action configWASVersion_number
   -remote_acl_user pdpermadmin
   -sec_master_pwd myPassword
   -pdmgrd_host fully_qualified_DN_of_the_policy_server_host
   -pdacld_host fully_qualified_DN_of_the_authorization_server_host
   -was_home c:\WebSphere\AppServer
   ```

   **Note:** The value of the -was_home option in the command above is shown as an example only. This value will change depending on the version of
WebSphere Application Server you are running and the platform you are using. For instance this value will be:

**Windows**

WebSphere Application Server version 4.0.6:
```
c:\WebSphere\AppServer
```

WebSphere Application Server version 5.0.2:
```
"c:\Program Files\WebSphere\AppServer"
```

**Solaris, Linux, HP-UX**
```
/opt/WebSphere/AppServer
```

**AIX**
```
/usr/WebSphere/AppServer
```

The `pdwascfg` utility configures WebSphere Application Server to use Tivoli Access Manager for WebSphere as the authorization vendor.

**Notes:**

1. The `pdwascfg` utility only supports domains with the administration user created as sec_master.

2. The `pdwascfg` utility creates a log file called `AMWASConfig.log` on the directory where the utility was run.

7. Verify that the `pdwascfg` command successfully created the PdPerm properties file.
   - Solaris, Linux, HP-UX
     ```
     /opt/WebSphere/AppServer/java/jre/PdPerm.properties
     ```
   - AIX
     ```
     /usr/WebSphere/AppServer/java/jre/PdPerm.properties
     ```
   - Windows
     - WebSphere Application Server version 4.0.6
       ```
       C:\WebSphere\AppServer\java\jre\PdPerm.properties
       ```
     - WebSphere Application Server version 5.0.2
       ```
       C:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties
       ```

**Note:** The above path names assume the default installation directory for WebSphere Application Server. If you installed in a non-default location, adjust the path names accordingly.

### Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6

If using WebSphere 5.0.2, skip this step and proceed to "Part 5b: Migrate WebSphere security settings — WebSphere version 5.0.2" on page 34.

This step migrates application security policy from the WebSphere `admin.ear` deployment descriptor file to the Tivoli Access Manager policy database. The migration utility creates in the Tivoli Access Manager object space, objects that represent WebSphere resources. You will not be able to start WebSphere if this section is not complete.

Complete the following steps:

1. Stop WebSphere if it is running.
2. Ensure that the WAS_HOME environment variable is set to the location of your WebSphere Application Server installation. The following examples show the default location:
   - Solaris, Linux, HP-UX
     WAS_HOME=/opt/WebSphere/AppServer
   - AIX
     WAS_HOME=/usr/WebSphere/AppServer
   - Windows
     WAS_HOME=C:\WebSphere\AppServer

3. Assemble the following information, which you will need to specify as input parameters to the migration utility:
   - The name of the EAR file to migrate. For this initial use of the migration utility, you must migrate the administration EAR file:
     - Solaris, Linux, HP-UX
       /opt/WebSphere/AppServer/config/admin.ear
     - AIX
       /usr/WebSphere/AppServer/config/admin.ear
     - Windows
       C:\WebSphere\AppServer\config\admin.ear
   - The full path to the PDPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.
     
     **Note:** The file location must be expressed as a Uniform Resource Identifier.
     - Solaris, Linux, HP-UX
       file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
     - AIX
       file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
     - Windows
       file:/c:\WebSphere\AppServer\java\jre\PdPerm.properties
   - The name of the Tivoli Access Manager administration account. This should be sec_master.
   - The password for the sec_master account.
   - The name of the WebSphere administrative user account. This should match the account you created/imported above. For example:
     wsadmin
   - The LDAP distinguished name (DN) suffix under which both the Tivoli Access Manager policy server and WebSphere Application Server store user information. This should match the DN suffix used when you created the wsadmin user.

The example shown in “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26 created wsadmin with following DN:

```
cn=wsadmin,o=ibm,c=us
```

In this case the DN suffix is: o=ibm,c=us

This value should be given as the argument to the –d option to the migrateEAR4 utility.
Note: You can use pdadmin to display the DN for wsadmin on your system:

```
padmin> user show wsadmin
```

4. Change directory to the location of the migration utility:

- (UNIX) /opt/amwas/bin
- (Windows) C:\Program Files\Tivoli\amwas\bin

5. Run the migration utility to migrate the data contained in admin.EAR.
Using the parameters that you assembled in the previous step, enter the following text at a command prompt, as one continuous command line:

**UNIX**

```
migrateEAR4 -j /opt/WebSphere/AppServer/config/admin.ear
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
```

Note that the default location of the PdPerm.properties file on AIX is:

```
/usr/WebSphere/AppServer/java/jre/PdPerm.properties
```

**Windows**

```
migrateEAR4 -j c:\WebSphere\AppServer\config\admin.ear
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:c:\WebSphere\AppServer\java\jre\PdPerm.properties
```

A status message is displayed upon completion of the migration. Output of the utility is logged to the file pdwas_migrate.log created on the directory where the utility was run. Check the log file to ensure that all policy was migrated for the application. If the log file displays errors, check the last transaction that occurred and correct the source of the error and re-run the migration tool.

If the migration was unsuccessful, verify that you supplied the correct Uniform Resource Identifier to the -c option, and the correct filename to the -j option.

The migration utility requires access to admin.ear. By default, the application assembly tool contains URL references to the location of the Document Type Definitions (DTD) standard. Thus, lookups for the deployment descriptor DTDs require a connection to the Internet. If the host computer is not connected to the Internet, use a local copy of the DTD. In this case, update the deployment descriptors to point to the local DTD.

**Attention:** You will need to run the migration utility at least one more time before using Tivoli Access Manager for WebSphere. You will need to run it against the EAR file for each application that you are securing. The instructions for doing this are located in Chapter 4, “Migrating security roles,” on page 43.

**Add the pdwas-admin group to the administration ACL**

Complete the following steps to add the pdwas-admin group to the administration ACL:

1. Use pdadmin to add the pdwas-admin group to the appropriate ACL. Enter the following text as one continuous command:

```
padmin> acl modify _WebAppServer_deployedResources_AdminRole_admin_ACL
set group pdwas-admin [WebAppServer]
```

2. If your secure domain contains more than one authorization server, use pdadmin to perform the server replicate command to ensure that all authorization servers are immediately updated with the ACL changes.
Part 5b: Migrate WebSphere security settings — WebSphere version 5.0.2

If you are using WebSphere Application Server 4.0.6, skip this step.

This step migrates application security policy from the WebSphere adminconsole.ear deployment descriptor file to the Tivoli Access Manager policy database. The migration utility creates in the Tivoli Access Manager object space, objects that represent WebSphere resources.

Note: Tivoli Access Manager for WebSphere does not support security for WebSphere Application Server administration tasks.

Complete the following steps:
1. Stop WebSphere if it is running.
2. Ensure that the WAS_HOME environment variable is set to the location of your WebSphere Application Server installation. The following examples show the default location:
   - Solaris, Linux, HP-UX
     WAS_HOME=/opt/WebSphere/AppServer
   - AIX
     WAS_HOME=/usr/WebSphere/AppServer
   - Windows
     WAS_HOME=C:\Program Files\WebSphere\AppServer
3. Assemble the following information, which you will need to specify as input parameters to the migration utility:
   - The name of the EAR file to migrate. For this initial use of the migration utility, you must migrate the administration EAR, admin-authz.xml and naming-authz.xml:
     - Solaris, Linux, HP-UX
       /opt/WebSphere/AppServer/installedApps/cellname/adminconsole.ear
       /opt/WebSphere/AppServer/config/cells/cellname/admin-authz.xml
       /opt/WebSphere/AppServer/config/cells/cellname/naming-authz.xml
     - AIX
       /usr/WebSphere/AppServer/installedApps/cellname/adminconsole.ear
       /usr/WebSphere/AppServer/config/cells/cellname/admin-authz.xml
       /usr/WebSphere/AppServer/config/cells/cellname/naming-authz.xml
     - Windows
       C:\Program Files\WebSphere\AppServer\installedApps\cellname\adminconsole.ear
       C:\Program Files\WebSphere\AppServer\config\cells\cellname\admin-authz.xml
       C:\Program Files\WebSphere\AppServer\config\cells\cellname\naming-authz.xml
   - The full path to the PDPPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.
     Note: The file location must be expressed as a Uniform Resource Identifier.
     - Solaris, Linux, HP-UX
       file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
     - AIX
       file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
- Windows
  file://"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"
- The name of the Tivoli Access Manager administration account. This should be sec_master.
- The password for the sec_master account.
- The name of the WebSphere administrative user account. This should match the account you created above. For example:
  wsadmin
- The LDAP distinguished name (DN) suffix under which both the Tivoli Access Manager policy server and WebSphere Application Server store user information. This should match the DN suffix used when you created the wsadmin user.

The example shown in "Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server" on page 26 created wsadmin with following DN:
  cn=wsadmin,o=ibm,c=us

In this case the DN suffix is: o=ibm,c=us

This value should be given as the argument to the -d option to the migrateEAR5 utility.

Note: You can use pdadmin to display the DN for wsadmin on your system:
  pdadmin> user show wsadmin

4. Change directory to the location of the migration utility:
- (UNIX) /opt/amwas/bin
- (Windows) C:\Program Files\Tivoli\amwas\bin

5. Run the migration utility to migrate the data contained in the files adminconsole.EAR, admin-authz.xml and naming-authz.xml.

Using the parameters that you assembled in the previous step, enter the following text at a command prompt, as one continuous command:

**Solaris, Linux, HP-UX**

```
  migrateEAR5
  -j /opt/WebSphere/AppServer/installedApps/cellname/adminconsole.ear
  -a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
  -c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
  -e adminconsole
  migrateEAR5
  -j /opt/WebSphere/AppServer/config/cells/cellname/admin-authz.xml
  -a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
  -c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
  migrateEAR5
  -j /opt/WebSphere/AppServer/config/cells/cellname/naming-authz.xml
  -a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
  -c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
```

**AIX**

```
  migrateEAR5
  -j /usr/WebSphere/AppServer/installedApps/cellname/adminconsole.ear
  -a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
  -c file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
  -e adminconsole
```
migrateEAR5
-j /usr/WebSphere/AppServer/config/cells/cellname/admin-authz.xml
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties

migrateEAR5
-j /opt/WebSphere/AppServer/config/cells/cellname/naming-authz.xml
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties

Windows
migrateEAR5
-j "c:\Program Files\WebSphere\AppServer\installedApps\cellname\adminconsole.ear"
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"
-e adminconsole

migrateEAR5
-j "c:\Program Files\WebSphere\AppServer\config\cells\cellname\admin-authz.xml"
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

migrateEAR5
-j "c:\Program Files\WebSphere\AppServer\config\cells\cellname\naming-authz.xml"
-a sec_master -p sec_master_password -w wsadmin -d "o=ibm,c=us"
-c file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

A status message is displayed upon completion of the migration. Output of the utility is logged to the file pdwas_migrate.log which is created on the directory where the utility was run. Check the log file to ensure that all policy was migrated for the application. If the log file displays errors, check the last transaction that occurred and correct the source of the error and re-run the migration tool.

If the migration was unsuccessful, verify that you supplied the correct Uniform Resource Indicator to the -c option, and the correct filename to the -j option.

The migration utility requires access to adminconsole.ear. By default, the application assembly tool contains URL references to the location of the Document Type Definitions (DTD) standard. Thus, lookups for the deployment descriptor DTDs require a connection to the Internet. If the host computer is not connected to the Internet, use a local copy of the DTD. In this case, update the deployment descriptors to point to the local DTD.

Attention: You will need to run the migration utility at least one more time before using Tivoli Access Manager for WebSphere. You will need to run it against the EAR file for each application that you are securing. Instructions for doing this are located in Chapter 4, “Migrating security roles,” on page 43.

Configuring Tivoli Access Manager for WebSphere in a WebSphere Application Server version 5.1 environment

There is no need to install either the Access Manager Java Runtime Environment or Tivoli Access Manager for WebSphere if they are being configured against a WebSphere Application Server version 5.1 installation. Both the Access Manager Java Runtime Environment and Tivoli Access Manager for WebSphere are shipped
as part of the WebSphere Application Server 5.1 package. However, the
configuration steps are different to other versions of WebSphere. Follow the
instructions in the sections below to configure Tivoli Access Manager for
WebSphere in a WebSphere Application Server version 5.1 environment.

Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server

If it doesn’t already exist, the WebSphere administrative user must be created in
Tivoli Access Manager prior to policy migration:

```
pdadmin -a sec_master -p sec_master_password
pdadmin> user create was_admin_uid was_admin_user_dn
was_admin_uid was_admin_uid was_admin_pwd
pdadmin> user modify was_admin_uid account-valid true
```

Part 2: Enable Security in WebSphere Application Server version 5.1

The steps for enabling security using Tivoli Access Manager for WebSphere
Application Server are identical to enabling native WebSphere Application Server
security. The important points to remember are:

- Tivoli Access Manager and WebSphere are sharing the same user registry.
  Therefore, WebSphere needs to be configured to use the same user registry as
  Tivoli Access Manager.
- When configuring the LDAP directory in the WebSphere Administration
  Console, you need to ensure the Use Tivoli Access Manager for Account Policies
  checkbox is ticked.

Part 3: Configure the Access Manager Java Runtime Environment

Refer to the "Configuring WebSphere Application Server to use Tivoli Access Manager for
authentication" section in the WebSphere Application Server version 5.1 InfoCenter
documentation for information on the required configuration steps.

Part 4: Configure Tivoli Access Manager for WebSphere

Perform the following steps to configure Tivoli Access Manager for WebSphere to
operate with WebSphere version 5.1.

1. Run the setupcmdline script located in \ WAS_HOME\bin to setup the environment.
2. Set the PDWAS_HOME environment variable to the value of the WAS_HOME
   environment variable. On Windows the command would be:
   ```
   set PDWAS_HOME=%WAS_HOME%
   ```
3. Run the pdwascfg script located in the %WAS_HOME%\bin directory to perform the
   configuration. The following example uses pdwascfg.bat. Replace this with
   pdwascfg.sh for UNIX environments:
   ```
   %WAS_HOME%\bin\pdwascfg.bat -action configWAS5
   -remote_acl_user remote_ACL_user_name
   -sec_master_pwd sec_master_pwd -pdmgrd_host TAM_Policy_Server_host
   -pdacld_host TAM_Authorization_Server_host -was_home WAS_HOME
   -amwas_home WAS_HOME -embedded true -action_type local -verbose true
   ```

The remote_ACL_user_name corresponds to the user created by the
configuration. This user is employed for all communication with the Tivoli
Access Manager servers. This is a special user that should not be used for any
other purpose.
Part 5: Migrate administration policy

In WebSphere Application Server version 5.1 extra console policy definition files need to be migrated to Tivoli Access Manager. The migration utility is located in the %WAS_HOME%/bin directory.

To migrate all the required policy (on Windows) the following commands need to be run (as one continuous line):

**UNIX**

```
Note: The default location of WebSphere on AIX is,  
/usr/WebSphere/AppServer
migrateEAR5  
-j /opt/WebSphere/AppServer/installedApps/cellname/adminconsole.ear  
-a sec_master -p sec_master_pwd  
-w was_admin_uid -e "adminconsole"  
-d "o=ibm,c=us"  
-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties  
-e adminconsole

migrateEAR5  
-j /opt/WebSphere/AppServer/config/cells/cellname/admin-authz.xml  
-a sec_master -p sec_master_pwd  
-w was_admin_uid -d "o=ibm,c=us"  
-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties

migrateEAR5  
-j /opt/WebSphere/AppServer/config/cells/cellname/naming-authz.xml  
-a sec_master -p sec_master_pwd  
-w was_admin_uid -d "o=ibm,c=us"  
-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
```

**Windows**

```
migrateEAR5  
-j "c:\Program Files\WebSphere\AppServer\installedApps\  
cellname\adminconsole.ear  
-a sec_master -p sec_master_pwd  
-w was_admin_uid  
-d "o=ibm,c=us" -c file:"c:\Program Files\WebSphere\AppServer\  
java\jre\PdPerm.properties"  
-e adminconsole

migrateEAR5  
-j "c:\Program Files\WebSphere\AppServer\config\cells\  
cellname\admin-authz.xml"  
-a sec_master -p sec_master_pwd  
-w was_admin_uid -d "o=ibm,c=us"  
-c file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

migrateEAR5  
-j "c:\Program Files\WebSphere\AppServer\config\cells\  
cellname\naming-authz.xml"  
-a sec_master -p sec_master_pwd  
-w was_admin_uid -d "o=ibm,c=us"  
-c file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"
```

The -e option is required for the migration of the adminconsole.ear file because WebSphere Application Server renames this application during deployment.
Configuring additional installations

This section describes how to configure additional Tivoli Access Manager for WebSphere installations into a Tivoli Access Manager secure domain.

The instructions in this section assume the following:

- You have successfully completed the instructions in “Configuring the initial installation” on page 25.
  
  By completing the above instructions, you will have previously migrated security information from the admin.ear file to Tivoli Access Manager if using WebSphere Application Server version 4.0.6 or the adminconsole.ear file to Tivoli Access Manager if using WebSphere Application Server version 5.0.2.

- You have installed Tivoli Access Manager for WebSphere on a different (additional) host system from the initial host system that you previously configured. You are now ready to configure Tivoli Access Manager for WebSphere on that additional host system.

**Note:** Do not use the instructions in this section unless you have previously completed the section “Configuring the initial installation” on page 25.

These instructions do not describe how to migrate security information from additional EAR files. You can complete the migration of any additional EAR files separate from completing the configuration instructions in this section. For more information on migrating EAR files, see Chapter 4, “Migrating security roles,” on page 43.

The configuration steps are summarized in the following diagram:

![Diagram](image)

*Figure 6. Configuration tasks for additional Tivoli Access Manager for WebSphere systems*

The configuration steps are described in the following sections:

- “Part A-1: Configure the Access Manager Java Runtime Environment”
- “Part A-2: Join a secure domain” on page 40

**Part A-1: Configure the Access Manager Java Runtime Environment**

Configure the Access Manager Java Runtime Environment component to access the Java runtime that is distributed with IBM WebSphere Application Server.

**Note:** The Access Manager Java Runtime Environment is a software prerequisite for Tivoli Access Manager for WebSphere.

The Access Manager Java Runtime Environment can be configured using either the Tivoli Access Manager Base configuration GUI or from the command line using the
**pdjrtecfg** command. To configure the Access Manager Java Runtime Environment from the Access Manager Base configuration GUI:

1. Change directory to the following location:
   - (UNIX) /opt/PolicyDirector/bin
   - (Windows) C:\Program Files\Tivoli\Policy Director\bin

2. Enter the following command:
   pdconfig

   The Access Manager Configuration screen is displayed from which you can configure the Java runtime.

To configure the Access Manager Java Runtime Environment component from the command line:

1. Verify that environment variable WAS_HOME is set to the IBM WebSphere Application Server home directory.

2. Change directory to the following location:
   - (UNIX) /opt/PolicyDirector/sbin
   - (Windows) C:\Program Files\Tivoli\Policy Director\sbin

3. Enter the following command:
   - (UNIX) pdjrtecfg -action config -java_home $WAS_HOME/java/jre
   - (Windows) pdjrtecfg -action config -java_home %WAS_HOME%\java\jre

   **Note:** Ensure that the location of the java binary that appears first in your PATH variable matches the location of the java binary that you specify to the **pdjrtecfg** option -java_home pathname.

**Part A-2: Join a secure domain**

Complete the following steps:

1. Stop the WebSphere Application Server.

2. Assemble the following information:
   - The name of the user account you want to use as a user identity for the Tivoli Access Manager for WebSphere application. The example commands in these instructions use the identity pdperm2admin. You can choose any name you want.

   **Note:** You can use an existing identity with the Tivoli Access Manager secure domain, or you can create a new identity. In most cases, you will create a new, unique identity to represent the Tivoli Access Manager for WebSphere component on the host system that you are currently configuring.
   - The password for the sec_master account.
   - The fully qualified domain name for the computer that hosts the policy server. For example: pdmgrsrver.mysubnet.ibm.com
   - The fully qualified domain name for the computer that hosts the authorization server. For example: pdacldserver.mysubnet.ibm.com

3. To set the WAS_HOME environment variable to the WebSphere Application Server installation directory, change directory to **WebSphere_install_directory/bin** and run the following command:

   **UNIX**

   setupCmdLine.sh
4. On UNIX platforms, set the PDWAS_HOME environment variable to the Tivoli Access Manager for WebSphere installation directory. On Windows platforms, PDWAS_HOME will already exist in the environment.

UNIX

PDWAS_HOME=/opt/amwas
export PDWAS_HOME

5. Change directory to:
   - UNIX: /opt/amwas/bin
   - Windows: C:\Program Files\Tivoli\amwas\sbin

6. Using the example parameters assembled previously, enter the following command, as one continuous command line, using either the `-action configWAS4` or `configWAS5` parameter depending on the version of WebSphere Application Server that you are using:

```
pdwascfg -action configWASversion_number
   -remote_acl_user pdperm2admin
   -sec_master_pwd myPassword
   -pdmgrd_host pdmgrserver.mysubnet.ibm.com
   -pdacld_host pdacldserver.mysubnet.ibm.com
   -was_home C:\WebSphere\AppServer
   [-amwas_home location_of_the_amwas_installation]
```

7. Verify that the `pdwascfg` command successfully created the PdPerm properties file.
   - Solaris, Linux, HP-UX
     /opt/WebSphere/AppServer/java/jre/PdPerm.properties
   - AIX
     /usr/WebSphere/AppServer/java/jre/PdPerm.properties
   - Windows
     - WebSphere Application Server version 4.0.6
       C:\WebSphere\AppServer\java\jre\PdPerm.properties
     - WebSphere Application Server version 5.0.2
       C:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties

**Note:** The above path names assume the default installation directory for WebSphere Application Server. If you installed in a non-default location, adjust the path names accordingly.
Chapter 4. Migrating security roles

Tivoli Access Manager for WebSphere provides a migration utility that automatically converts security role definitions to Tivoli Access Manager protected objects. The role definitions are read from the WebSphere application deployment descriptors and migrated to the Tivoli Access Manager protected object space. This chapter describes how to use the utility.

Topic index:
- “How to migrate security roles”
- “Migration utility limitations” on page 47
- “Troubleshooting tips” on page 47

How to migrate security roles

These instructions are intended to be used after completion of an initial configuration of Tivoli Access Manager for WebSphere as discussed in Chapter 3, “Configuration procedures,” on page 25.

To migrate J2EE application security roles to Tivoli Access Manager for WebSphere, complete the following steps:

1. Verify that you are logged in as root on UNIX systems or as a user with administrative privileges on Windows systems.

2. The migration utility requires access to the deployment descriptors for the applications that have been secured. By default, the application assembly tool contains URL references to the location of the Document Type Definitions (DTD) standard. Thus, lookups for the deployment descriptor DTDs require a connection to the Internet. If the host computer is not connected to the Internet, use a local copy of the DTD. In this case, update the deployment descriptors to point to the local DTD.

3. To set the WAS_HOME environment variable to the WebSphere Application Server installation directory, change directory to WebSphere_install_directory/bin and run the following command:

   UNIX
   ```bash
   setupCmdLine.sh
   ```

   Windows
   ```dos
   setupCmdLine.bat
   ```

4. On UNIX platforms, set the PDWAS_HOME environment variable to the Tivoli Access Manager for WebSphere installation directory. On Windows platforms, PDWAS_HOME will already exist in the environment.

   UNIX
   ```bash
   PDWAS_HOME=/opt/amwas
   export PDWAS_HOME
   ```

5. Assemble the following information, which you will need to specify as input parameters to the migration utility:

   - The name of the EAR file to migrate. For example:
     - Solaris, Linux, HP-UX
     - WebSphere Application Server version 4.0.6:
The location of the PdPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.

**Note:** The file location must be expressed as a Uniform Resource Indicator.

- **Solaris, Linux, HP-UX**
  
  file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties

- **AIX**
  
  file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties

- **Windows**
  
  - WebSphere Application Server version 4.0.6:
    
    file:/c:\WebSphere\AppServer\installedApps\secureApp.ear
  
  - WebSphere Application Server version 5.0.2:
    
    file:/c:\Program Files\WebSphere\AppServer\installedApps\cellname\secureApp.ear

- The location of the PdPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.

  **Note:** The file location must be expressed as a Uniform Resource Indicator.

  - **Solaris, Linux, HP-UX**
    
    file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
  
  - **AIX**
    
    file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
  
  - **Windows**
    
    - WebSphere Application Server version 4.0.6:
      
      file:/c:\WebSphere\AppServer\java/jre/PdPerm.properties
    
    - WebSphere Application Server version 5.0.2:
      
      file:/c:\Program Files\WebSphere\AppServer\java/jre/PdPerm.properties

  - The name of the Tivoli Access Manager administration account. This should be sec_master.
  
  - The password for the sec_master account.
  
  - The name of the WebSphere administrative user account. This should match the account you created during the initial configuration of Tivoli Access Manager for WebSphere. For example:
    
    wsadmin
  
  - The LDAP distinguished name (DN) suffix under which both the Tivoli Access Manager policy server and WebSphere Application Server store user information. This should match the DN suffix used when you created the wsadmin user.

  **The example shown in “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26 created wsadmin with following DN:**

  cn=wsadmin,o=ibm,c=us

  In this case the DN suffix is: o=ibm,c=us

  This value should be given as the argument to the –d option to the migrateEAR utility.
Note: You can use `pdadmin` to display the DN for `wsadmin` on your system:

```
padmin> user show wsadmin
```

- The application display name. An application name can be changed at application deployment, or later through the WebSphere console. This change will not be reflected in the EAR file. When the EAR file is not modified to reflect the new name, the wrong protected objects will be created. Use the `-e` option to specify the name of the application as it is displayed on the WebSphere Application Server console.

6. Ensure that you have the most recent EAR file for the application. Ensure that the EAR file has all the expected user to role mappings. If you are uncertain if all role mappings are present, export the application.

   See the IBM WebSphere Application Server documentation for instructions on exporting EAR files.

7. Change directory to the location of the migration utility:
   - (UNIX) `/opt/amwas/bin`
   - (Windows) `C:\Program Files\Tivoli\amwas\bin`

8. Run the migration utility to migrate the application data.

   Using the parameters that you assembled in the previous step, enter the following at a command prompt, as one continuous command line:

   **For WebSphere Application Server version 4.0.6:**

   ```
   UNIX
   migrateEAR4 
   -j /opt/WebSphere/AppServer/installedApps/your_application.ear
   -a sec_master
   -p sec_master_password
   -w wsadmin
   -d "o=ibm,c=us"
   -c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties 
   [-e application_name]
   
   Windows
   migrateEAR4 -j \WebSphere\AppServer\installedApps\your_application.ear
   -a sec_master
   -p sec_master_password
   -w wsadmin
   -d "o=ibm,c=us"
   -c file:\WebSphere\AppServer\java\jre\PdPerm.properties 
   [-e application_name]
   ```
For WebSphere Application Server version 5.0.2:

### UNIX

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrateEAR5</td>
</tr>
<tr>
<td>-j /opt/WebSphere/AppServer/installedApps/cellname/your_application.ear</td>
</tr>
<tr>
<td>-a sec_master</td>
</tr>
<tr>
<td>-p sec_master_password</td>
</tr>
<tr>
<td>-w wasadmin</td>
</tr>
<tr>
<td>-d &quot;o=ibm,c=us&quot;</td>
</tr>
<tr>
<td>-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties [-e application_name]</td>
</tr>
</tbody>
</table>

Note that the default location of the PdPerm.properties file on AIX is:

/usr/WebSphere/AppServer/java/jre/PdPerm.properties

### Windows

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrateEAR5</td>
</tr>
<tr>
<td>-j &quot;c:\Program Files\WebSphere\AppServer\installedApps\cellname\your_application.ear&quot;</td>
</tr>
<tr>
<td>-a sec_master</td>
</tr>
<tr>
<td>-p sec_master_password</td>
</tr>
<tr>
<td>-w wasadmin</td>
</tr>
<tr>
<td>-d &quot;o=ibm,c=us&quot;</td>
</tr>
<tr>
<td>-c file:c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties [-e application_name]</td>
</tr>
</tbody>
</table>

A status message is displayed upon completion of the migration. Output of the utility is logged to the file pdwas_migrate.log which is created on the directory where the utility was run. Check the log file to ensure that all policy was migrated for the application. If the log file displays errors, check the last transaction that occurred and correct the source of the error and re-run the migration tool.

If the migration was unsuccessful, verify that you supplied the correct Uniform Resource Indicator to the -c option, and the correct filename to the -j option.

9. Repeat the previous steps for each Enterprise Archive (EAR) file that contains role definitions that must be migrated to Tivoli Access Manager.

There is no need to run the migration utility against J2EE applications that do not have security information in their deployment descriptors.

**Note:** Run the migration utility only once for each unique EAR file. When there are multiple copies of any EAR file, you do not need to run the migration utility for each copy. The migration utility needs to be run only once per Tivoli Access Manager domain.

10. Choose one of the following actions:

    • If you are using WebSphere Application Server Advanced Edition Single Server, go to the next step.

    • If you are not using WebSphere Application Server Single Server edition, the migration is complete. Do not perform the next step.

11. When using Tivoli Access Manager for WebSphere with WebSphere Single Server Edition, you must use padmin to manually add users to the ACLs that the migration utility created. Example padmin commands to add users are described in "Migration utility limitations” on page 47.

You can also review how to add users to ACLs in the sample application described in the tutorial, Chapter 6, “Tutorial: How to enable security,” on page 69. See the example commands in the section Migrate the application to Tivoli Access Manager.
Migration utility limitations

The migration utility has the following limitations:

- The migration utility is designed only to migrate the roles in EAR files to the Tivoli Access Manager protected object space. Do not use the migration utility as a maintenance utility for roles. After migrating an EAR file, use either the Web Portal Manager or the **pdadmin** utility to manage roles.

- The migration utility migrates only the user and roles specified in the EAR file. Ensure that you are using the latest EAR file for your application.

- After the migration utility has been run once against an EAR file, it is recommended that you do not run it again when changes are made to an EAR file. The following problems can occur when an EAR is created and migrated to the protected object space, and then is migrated again.
  - On the second or subsequent migrations, if an existing role has been removed from the EAR, it will not be removed from the protected object space.
  - On the second or subsequent migrations, changes to the EAR file might require the migration utility to instruct Tivoli Access Manager to delete an ACL definition. In some cases, Tivoli Access Manager may prevent this deletion. Note that the migration of an EAR file to the Tivoli Access Manager protected object space results in the creation of ACLs that are attached to objects. If the administrator has manually attached the ACL definition to other protected objects, Tivoli Access Manager prevents removal of the ACL. Thus, even if the original object that was created by the first run of the migration utility no longer exists, the ACL cannot be deleted.

- Use **pdadmin** to modify roles. You can use **pdadmin** to add additional roles.

- When using the migration utility with WebSphere Application Server Advanced Edition Single System Edition, you must manually add the users to the ACLs that were created by the migration utility. This limitation does not affect WebSphere Application Server Advanced Edition.

Use **pdadmin** to add the users to the ACL. The following example shows how to add users to an ACL, based on the sample application described in section "Part 8: Migrate the application to Tivoli Access Manager" on page 75 in the tutorial chapter. Note that each **pdadmin** command must be entered as one continuous command line.

```plaintext
C:> pdadmin -a sec_master -p myPassword
pdadmin> acl list
(Find the ACL that starts with _WebAppServer_deployedResources_GoodGuys_)

pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_simpleSessionApp_ACL
add user user1 T[WebAppServer]

pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_simpleSessionApp_ACL
add user user2 T[WebAppServer]

pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_simpleSessionApp_ACL
add user user3 T[WebAppServer]

pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_simpleSessionApp_ACL
add user user4 T[WebAppServer]

pdadmin> exit
```

Troubleshooting tips

This section contains the following topics:

- “Use of log files” on page 48
- “Users not attached to created ACLs” on page 48
- “Migration fails on Windows files with short name” on page 48
- “Web Portal Manager unable to attach an ACL to an object” on page 48
Use of log files

When troubleshooting problems with the migration utility, use the log files that WebSphere and Tivoli Access Manager provide:

- Configure logging for the Tivoli Access Manager authorization server. Difficulties with accessing objects in the protected object namespace are logged here. Note that log information generated by requests from the Tivoli Access Manager authorization component is logged here. Note also that this log is not the same as the WebSphere logs. For more information, see the IBM Tivoli Access Manager Base Administration Guide.
- Activity by the migration utility is logged in the file pdwas_migrate.log. This file is located in the directory where the migration utility is run. The last log message normally describes what the migration utility was attempting to do most recently. Therefore, in most cases it will indicate where an error was generated.

Users not attached to created ACLs

Problem: The admin.ear file does not contain any user information, just role mappings. As a result, no users are attached to the created ACLs.

Solution: Use pdadmin to add the group pdwas-admin to the ACL. Enter the following command as one continuous command line:

```
pdadmin> acl modify _WebAppServer_deployedResources_AdminRole_admin_ACL
set group pdwas-admin T[WebAppServer]
```

Migration fails on Windows files with short name

Problem: The migration utility does not work on filenames containing a tilde (~). This can cause problems when attempting to migrate a Windows short file name.

Solution: Rename the filenames to omit the tilde (~)

Web Portal Manager unable to attach an ACL to an object

Problem: The Web Portal Manager might be unable to attach an ACL to objects that contain spaces in the object name.

Workaround: As a workaround, use pdadmin to attach the ACL.

Solution: If possible, before running the migration utility, ensure that there are no spaces in the definitions listed in the deployment descriptors. Verify that the application name does not contain spaces.

Warning that user [...] is a member of pdwas-admin

Problem: When the migration utility is run, you might see a WARNING message, indicating that user wsadmin is a member of the group pdwas-admin.

Solution: This warning is expected and is displayed for security purposes only. The purpose of this warning is to identify the user as a current member of the pdwas-admin group, so that the administrator can verify the accuracy of the list of users contained in this important administration group.
Note: Members of the pdwas-admin group can be updated through the WebSphere administration console or the Tivoli Access Manager Web Portal Manager.

Client authentication lost due to session expiration

Problem: Tivoli Access Manager provides a default SSL timeout value for connections to the Tivoli Access Manager policy server. When this timeout value is exceeded during execution of the migration utility, you might see the following message:
The server lost the client’s authentication, probably because of session expiration.

Solution: When this message occurs, run the migration utility again using the -t minutes option. The migration utility uses a default of 60 minutes. This value should be no greater than the current SSL timeout between the authorization API client and the policy server.

You can determine the SSL timeout value by examining the parameter ssl-v3-timeout, located under the [ssl] stanza in the Tivoli Access Manager configuration file ivmgrd.conf. The default value for ssl-v3-timeout is 7200 seconds (120 minutes). When this default value is set, ensure that the SSL timeout set by the migration utility -t flag is at least 60 minutes.

For more information, see the IBM Tivoli Access Manager Base Administration Guide.

Migration utility messages not displayed in correct language

Problem: On Windows systems, messages from the Tivoli Access Manager for WebSphere migration utility are not displayed correctly for some languages, such as Brazilian Portuguese.

Workaround: Modify DOS Windows properties:
1. Enter the following command at a DOS command prompt:
   MS00S> chcp 1252
2. From the DOS window menu, select Properties.
   Note that Lucida Console is a True Type font.
4. Select OK. Select OK on the panel to apply the properties to the current window only.
5. You can now view the output from the migration utility.
Chapter 5. Administration tasks

This chapter contains the following topics:

- “WebSphere Advanced Edition Single Server version 4.0.6”
- “Tivoli Access Manager administration tools” on page 52
- “Specifying runtime properties” on page 52
- “Adding an object class to the console” on page 56
- “Configuring additional authorization servers” on page 55
- “GSO principal mapping setup” on page 57
- “Tivoli Access Manager for WebSphere logging” on page 61
- “Single Sign-On to WebSphere Application Server using WebSEAL” on page 63
- “Troubleshooting tips” on page 65
- “Backing up Tivoli Access Manager for WebSphere files” on page 67

WebSphere Advanced Edition Single Server version 4.0.6

IBM WebSphere Application Server provides a version of Advanced Edition that supports a single server. This version is designed to run WebSphere with host-based security instead of an external user registry.

This version of WebSphere Application Server is very useful for developing and prototyping applications and for demonstration of WebSphere Application Server features and capabilities. The system registry cannot be modified from the WebSphere console.

Tivoli Access Manager supports a number of external user registry types. When Tivoli Access Manager is used with WebSphere Advanced Edition Single Server, the Tivoli Access Manager administrator must create equivalent user registry entries for each relevant user account on the system that hosts WebSphere. This means that the user definitions in the user registry must be created manually.

Note that when users are mirrored into the Tivoli Access Manager user registry from the operating system entry, the Tivoli Access Manager user identity (ID) must match the operating system user ID. On Windows systems, this ID does not include the domain name.

Note also that the Tivoli Access Manager for WebSphere migration utility, when used with WebSphere Advanced Edition Single Server, does not automatically add users to access control lists that it creates. The administrator must manually add the users. For more information, see “Migration utility limitations” on page 47.

Use of Tivoli Access Manager for WebSphere with the WebSphere Advanced Edition Single Server is not recommended for production systems.

See “User registry prerequisites” on page 14.
Tivoli Access Manager administration tools

Do not use the WebSphere Application Server console to modify attributes for users or roles. These changes will not be reflected in the Tivoli Access Manager policy database.

All administration of user and role configuration information must be performed through one of the Tivoli Access Manager administration tools:
• The **pdadmin** command line utility
• The Tivoli Access Manager Web Portal Manager graphical user interface

Tivoli Access Manager also provides an administration API that can be used to perform administration tasks programmatically.

For more information on the Tivoli Access Manager administration tools, see the following guides:
• For **pdadmin** and for the graphical user interface, see the *IBM Tivoli Access Manager Base Administration Guide*.
• For the programmatic API, see the *IBM Tivoli Access Manager for e-business Administration C API Developer Reference* or *IBM Tivoli Access Manager for e-business Administration Java Classes Developer Reference*.

Specifying runtime properties

Tivoli Access Manager for WebSphere uses a Java property file that contains configuration parameters. The property file is created during the running of the **pdwascfg** utility and can be used subsequently to modify the configuration parameters.

The Java property file should be created in the following location:
• UNIX: `WAS_HOME/etc/PDWAS.properties`
• Windows: `WAS_HOME\etc\PDWAS.properties`

The following sections describe how to modify property settings:
• “Configuring static role caching”
• “Define static roles”
• “Configure dynamic role caching” on page 53
• “Role Based Policy Framework Parameters” on page 54

Configuring static role caching

Setting the static role cache
com.tivoli.pd.as.cache.StaticRoleCache=com.tivoli.pd.as.cache.StaticRoleCacheImpl

Enable static role caching
Enables or disables static role caching. Static role caching is enabled by default.
com.tivoli.pd.as.cache.EnableStaticRoleCaching=true

Define static roles
Defines additional static roles that are not defined in the WebSphere Application Server admin.ear or adminconsole.ear file (depending on the version of WebSphere that is running).
com.tivoli.pd.as.cache.StaticRoleCache.Roles=Administrator,Operator,Monitor,Deployer
Note: Application performance can be enhanced by adding the static roles: CosNamingRead, CosNamingWrite, CosNamingCreate, CosNamingDelete.

Configure dynamic role caching
This section describes the following settings:
- “Set the dynamic role cache”
- “Enable dynamic role caching”
- “Specify maximum number of users”
- “Specify principal lifetime”
- “Specify role lifetime”
- “Specify number of cache tables”

Set the dynamic role cache
com.tivoli.pd.as.cache.DynamicRoleCache=com.tivoli.pd.as.cache.DynamicRoleCacheImpl

Enable dynamic role caching
Enables or disables dynamic role caching. Dynamic role caching is enabled by default.
com.tivoli.pd.as.cache.EnableDynamicRoleCaching=true

Specify maximum number of users
The maximum number of users that the cache supports before a cache cleanup is performed. This parameter is used when dynamic role caching is enabled. The default number of users is 100000.
com.tivoli.pd.as.cache.DynamicRoleCache.MaxUsers=100000

Specify principal lifetime
The period of time in minutes that a principal entry is stored in the cache. This parameter is used when dynamic role caching is enabled. The default time is 10 minutes.
com.tivoli.pd.as.cache.DynamicRoleCache.PrincipalLifeTime=10

The term principal here refers to the Tivoli Access Manager credential returned from a unique LDAP user.

Specify role lifetime
The period of time in seconds that a role is stored in the role list for a user before it is discarded. This parameter is used when dynamic role caching is enabled. The default is 20 seconds.
com.tivoli.pd.as.cache.DynamicRoleCache.RoleLifetime=20

Specify number of cache tables
The number of tables used internally by the dynamic role cache. This parameter is used when dynamic role caching is enabled. The default is 20.

When a large number of threads use the cache, increase the value to tune and optimize cache performance.
com.tivoli.pd.as.cache.DynamicRoleCache.NumBuckets=20
Role Based Policy Framework Parameters

The Tivoli Access Manager for WebSphere role based policy framework parameters are set automatically by the `pdwascfg` utility at the time of Tivoli Access Manager for WebSphere configuration. It is very unlikely that you will need to change these parameters. The following list describes each parameter:

- **com.tivoli.pd.as.rbpf.AmasSession.AMGroup=amgroup-admin**
  
  Defines the group name for administrators that can traverse roles without being able to access those roles. This group of users is created to help with the administration of roles. The default value is `amgroup-admin`.

- **com.tivoli.pd.as.rbpf.AMAction=i**
  
  This parameter is used by the migration tool and the Tivoli Access Manager for WebSphere runtime to signify that a user is granted access to a role. This value is added to a Tivoli Access Manager ACL. It places invoke access on roles for users and groups.

- **com.tivoli.pd.as.rbpf.AMActionGroup=WebAppServer**
  
  This parameter is used by the migration tool and the Tivoli Access Manager for WebSphere runtime. It sets the Tivoli Access Manager action group that serves as a container for the action specified by `AMAAction` property.

- **com.tivoli.pd.as.rbpf.PosRoot=WebAppServer**
  
  This parameter is used by the migration tool and the Tivoli Access Manager for WebSphere runtime. It is used to determine where roles are stored in the protected object space.

- **com.tivoli.pd.as.rbpf.ProductId=deployedResources**
  
  This parameter is used by the migration tool and the Tivoli Access Manager for WebSphere runtime. It is used to determine where roles are stored in the protected object space. The default value is `deployedResources`.

- **com.tivoli.pd.as.rbpf.ResourceContainerName=Resources**
  
  This parameter is not used by Tivoli Access Manager for WebSphere but a value is required to be set. The default value is `Resources`.

- **com.tivoli.pd.as.rbpf.RoleContainerName=**
  
  This parameter is used by the Tivoli Access Manager for WebSphere runtime. It locates roles stored in the Role container name. It is used to determine where roles are stored in the protected object space. The default value is empty to support previous versions of the object space and the current implementation of the migration tool. Adding a value to this parameter will change the object space layout and any migrated applications will fail to be authorized.

- **com.tivoli.pd.as.rbpf.GrantUnprotectedAccess=true**
  
  This parameter is not used by Tivoli Access Manager for WebSphere, though it still needs to be set. The default value is `true`.

- **com.tivoli.pd.as.rbpf.UseEntitlements=false**
  
  Enables or disables the use of entitlement services by Tivoli Access Manager for WebSphere that are shipped with the WebSphere Application Server. Set to true, the entitlements service must be configured and working, and all ACLDs must be configured in the configuration URL. The default value is `false`.

- **com.tivoli.pd.as.rbpf.AmasSession.CfgURL=**
  
  The value of this property is configured based on the location of WebSphere and the `-cfg_url` specified when the `pdwascfg` utility was run.

- **com.tivoli.pd.as.rbpf.AmasSession.LoggingURL=**
  
  `file:/c:\WebSphere\AppServer\etc\jlog.properties`
The value of this property is configured based on the location of the Tivoli Access Manager for WebSphere installation at the time of configuration.

- `com.tivoli.pd.as.rbpf.AmasSession.AMName` =
  
  This value is set during Tivoli Access Manager for WebSphere configuration. It is the user specified in the `-remote_acl_user` parameter when the `pdwascfg` command was run.

---

**Configuring additional authorization servers**

Tivoli Access Manager secure domains can optionally contain more than one authorization server. The configuration of multiple authorization servers may be useful for two reasons:

- Failover capability, in case one authorization server is not available
- Performance improvement, when the volume of access requests is very large

Tivoli Access Manager for WebSphere can be configured to access multiple authorization servers. Use the Java class `com.tivoli.pd.jcfg.SvrSslCfg` to add additional authorization servers. The command syntax is:

```java
java com.tivoli.pd.jcfg.SvrSslCfg -action addsvr
-authsvr host_name:port_number:rank -cfg_file cfg_file
```

**Note:** Enter the above command as *one continuous command line.*

*Table 7. Command parameters for adding an authorization server*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-action addsvr</code></td>
<td>Add server information to the application server configuration file.</td>
</tr>
<tr>
<td><code>-authsvr</code></td>
<td>A Tivoli Access Manager authorization server. The argument format is:</td>
</tr>
<tr>
<td></td>
<td>- <code>host_name</code></td>
</tr>
<tr>
<td></td>
<td>A string. The name of the host computer for the authorization server</td>
</tr>
<tr>
<td></td>
<td>- <code>port_number</code></td>
</tr>
<tr>
<td></td>
<td>An integer value. The port on which the authorization server can be contacted.</td>
</tr>
<tr>
<td></td>
<td>- <code>rank</code></td>
</tr>
<tr>
<td></td>
<td>An integer value. The priority of this authorization server, relative to other authorization servers. Authorization servers with a higher rank will be contacted first when the application server is attempting to obtain an accept or deny decision for an access request. Failover occurs in order of rank.</td>
</tr>
</tbody>
</table>
Adding an object class to the console

The WebSphere Application Server console can be used to specify security policies for applications running in the WebSphere environment. The WebSphere Application Server console can also specify security policies for other Web resources, based on the entities stored in the user directory.

Tivoli Access Manager adds to the user registry the object class accessGroup. Tivoli Access Manager administrators can use the pdadmin command or the Web Portal Manager to create new groups. These new groups will be of object class accessGroup.

The WebSphere Application Server console is not configured by default to recognize objects of the class accessGroup as user registry groups. You can configure the WebSphere Application Server console to add this object class to the list of object classes that represent user registry groups.

Complete the following instructions:

1. Start the WebSphere Application Server if it is not already running.
2. From the WebSphere console, access the advanced settings for configuring security. For WebSphere Application Server version 4.0.6 the menu sequence is, Security → User Registries → LDAP → Advanced LDAP Settings.
3. Modify the Group Filter field. Add the following entry:
   (objectClass=accessGroup)

   For example, the Group Filter field would then look like:
   ((&(cn=%w)(|(objectclass=groupOfNames)
      (objectclass=groupOfUniqueNames)
      (objectclass=accessGroup)))

4. Modify the Group Member ID Map field. Add the following entry:
   accessGroup:member

   For example, the Group Member ID Map field would then look like:
5. Stop and restart WebSphere Application Server as instructed by the console.

GSO principal mapping setup

Tivoli Access Manager for WebSphere can be configured to manage authentication to WebSphere Enterprise Information Systems (EIS) such as databases, transaction processing systems and message queue systems, located within the WebSphere Application Server security domain. Authentication to the EIS security domain is achieved by Tivoli Access Manager for WebSphere using the GSO Principal Mapper JAAS login module for J2C resources. A special purpose login module inserts a credential into the JAAS Subject which the Resource Adapter can then use to authenticate to the backend EIS. The JAAS login module used is configured on a per-connection factory basis. The default WebSphere Application Server implementation of the principal mapping module retrieves the username and password information from XML configuration files. Tivoli Access Manager for WebSphere bypasses the credential stored in the XML configuration files and instead uses the Tivoli Access Manager GSO database to provide the EIS security domain authentication information.

WebSphere Application Server provides a default principal mapping module that associates user credential information with EIS resources. The default mapping module is defined in the WebSphere Application Server Administration Console from Security → JAAS Configuration → Application Logins. The mapping module name is DefaultPrincipalMapping. The user ID and password for the EIS security domain is defined under each connection factory by an authDataAlias attribute. The authDataAlias attribute does not actually contain the user name and password. An authDataAlias attribute contains an alias that refers to a user name and password pair that is defined in the security configuration document.

The Tivoli Access Manager Principal Mapping module processes the authDataAlias to determine the GSO resource name and user name required to perform the lookup on the Tivoli Access Manager GSO database. It communicates with the Tivoli Access Manager Policy Server which retrieves the GSO data from the registry.

Tivoli Access Manager stores authentication information on the Tivoli Access Manager GSO database against a resource/user name pair.
Creating a new application login

To create a new application login that uses the Tivoli Access Manager GSO database to store the login credentials:

1. Select Security → JAAS Configuration → Application Logins. Click the New button to create a new JAAS login configuration.
2. Enter the alias name of the new application login. Click Apply.
3. In the Additional Properties section, click the JAAS Login Modules link to define the JAAS Login Modules.
4. Click New and enter the JAAS Login Module:
   com.ibm.ws.security.common.auth.module.proxy.WSLoginModuleProxy
   Click Apply.
5. In the Additional Properties section, click Custom Properties to define Login Module-specific values which are passed directly to the underlying Login Modules.
6. Click New.

The Tivoli Access Manager principal mapping module uses the configuration string, authDataAlias, to retrieve the correct user name and password from the security configuration.

The authDataAlias passed to the module is configured for the J2CConnectionFactory. Since the authDataAlias is an arbitrary string entered at configuration time the following scenarios are possible:

---

Figure 7. GSO principal mapping architecture.
• The authDataAlias contains both the GSO Resource name and the user name. The format of this string is "Resource/User"
• The authDataAlias contains only the GSO Resource name. The user name is determined using the Subject of the current session.

Which scenario to use is determined by a JAAS configuration option. The details of these options are:

**Name:** com.tivoli.pd.as.gso.AliasContainsUserName

**Value:** True if the alias contains the user name, false if the user name should be retrieved from the security context.

When entering authDataAliases through the WebSphere Application Server console, the node name is automatically pre-pended to the alias. The JAAS configuration entry is to determine whether this node name should be removed or included as part of the resource name.

**Name:** com.tivoli.pd.as.gso.AliasContainsNodeName

**Value:** True if the alias contains the node name.

7. Enter each new parameter using the table below as a guideline.

<table>
<thead>
<tr>
<th>Scenario 1:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth Data Alias</td>
<td>BackendEIS/eisUser</td>
</tr>
<tr>
<td>Resource</td>
<td>BackEndEIS</td>
</tr>
<tr>
<td>User</td>
<td>eisUser</td>
</tr>
</tbody>
</table>

**Principal Mapping Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>true</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 2:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth Data Alias</td>
<td>BackendEIS</td>
</tr>
<tr>
<td>Resource</td>
<td>BackEndEIS</td>
</tr>
<tr>
<td>User</td>
<td>Currently authenticated WAS user</td>
</tr>
</tbody>
</table>

**Principal Mapping Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 3:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Auth Data Alias</td>
<td>nodename/BackendEIS/eisUser</td>
</tr>
<tr>
<td>Resource</td>
<td>BackEndEIS</td>
</tr>
<tr>
<td>User</td>
<td>eisUser</td>
</tr>
</tbody>
</table>

**Principal Mapping Parameters**
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapdelegateper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>true</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>true</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

**Scenario 4:**

Auth Data Alias      | nodename/BackendEIS/eisUser                        |
Resource              | nodename/BackendEIS (notice that node name was not stripped off) |
User                  | eisUser                                              |

**Principal Mapping Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>true</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

**Scenario 5:**

Auth Data Alias      | nodename/BackendEIS                        |
Resource              | BackEndEIS                       |
User                  | Currently authenticated WAS user         |

**Principal Mapping Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>true</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

**Scenario 6:**

Auth Data Alias      | nodename/BackendEIS/eisUser                        |
Resource              | nodename/BackendEIS/eisUser (notice that the Resource is the same as Auth Data Alias). |
User                  | Currently authenticated WAS user                  |

**Principal Mapping Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegate</td>
<td>com.tivoli.pd.as.gso.AMPrincipalMapper</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsUserName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.aliasContainsNodeName</td>
<td>false</td>
</tr>
<tr>
<td>com.tivoli.pd.as.gso.AMLoggingURL</td>
<td>file:///&lt;jlog.props.path&gt;</td>
</tr>
</tbody>
</table>

You now need to create the J2C authentication aliases. The user name and password assigned to these alias entries is irrelevant as Tivoli Access Manager is responsible for providing user names and passwords. However, the user name and password assigned to the J2C authentication aliases need to exist so they can be selected for the J2C connection factory in the console.
To create the J2C authentication aliases, from the WebSphere Application Server console, select Security → J2C Authentication Data and click the New button for each entry. Refer to the table above for scenario inputs.

The connection factories for each resource adapter that needs to use the GSO database must be configured to use the Tivoli Access Manager Principal Mapping module. To do this:
1. From the WebSphere Application Server console, select Applications → Enterprise Applications
2. Click on the application name.
3. From the Resource Adapter section towards the bottom of the application details screen, click the Connector Modules link.
4. Click on the .rar link.
5. Within the Additional Properties section towards the bottom of the screen, click on the Resource Adapter link.

Note: The resource adapter does not need to be packaged with the application. It can be standalone. For such a scenario the resource adapter is configured from Resources → Resource Adapters.
6. Within the Additional Properties section towards the bottom of the screen, click on the J2C Connection Factories link.
7. Click New and enter the connection factory properties.

The GSO Principal Mapper module is independent of the other Tivoli Access Manager for WebSphere functionality. However, it does require some of the files created during execution of pdwascfg, these being, JAVA_HOME/PdPerm.properties and PDWAS_HOME/etc/jlog.properties. To configure the GSO Principal Mapper module without configuring Tivoli Access Manager for WebSphere, the PdPerm.properties file needs to be created by manually invoking SvrSslCfg and the jlog.properties.template file needs to be manually copied to jlog.properties.

---

**Tivoli Access Manager for WebSphere logging**

The destination for the Tivoli Access Manager for WebSphere messages and trace logging is the WebSphere Application Server SystemOut.log file. This file is located in the $WAS_HOME/logs/cellname directory.

Tivoli Access Manager for WebSphere logging uses the JLog logging framework as does the Access Manager Java Runtime Environment. Tracing and messaging can be enabled selectively for various Tivoli Access Manager for WebSphere components.

Tracing and message logging for these components is controlled through an installed file named jlog.properties which can be found in the $AMWAS_HOME/etc directory.

The contents of this file lets the user control:

- Whether tracing is enabled or disabled for each of the Tivoli Access Manager for WebSphere components.
- Whether message logging is enabled or disabled for each of the Tivoli Access Manager for WebSphere components.
The `jlog.properties` file defines several "loggers", each of which is associated with one of the major Tivoli Access Manager for WebSphere components. These loggers include:

<table>
<thead>
<tr>
<th>Logger Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmasRBPFTraceLogger</td>
<td>Used to log messages/trace for the Role Based Policy Framework. This is an underlying framework used by Tivoli Access Manager for WebSphere to make access decisions.</td>
</tr>
<tr>
<td>AmasRBPFMessageLogger</td>
<td></td>
</tr>
<tr>
<td>AmasCacheTraceLogger</td>
<td>Used to log messages/trace for the policy caches defined by the Role Based Policy Framework.</td>
</tr>
<tr>
<td>AmasCacheMessageLogger</td>
<td></td>
</tr>
<tr>
<td>AMWASWebTraceLogger</td>
<td>Used to log messages/trace for the WebSphere Application Server authorization plug-in. For most problems only tracing for this component should need to be enabled.</td>
</tr>
<tr>
<td>AMWASWebMessageLogger</td>
<td></td>
</tr>
</tbody>
</table>

The implementation of these loggers routes messages to the WebSphere Application Server logging sub-system. Therefore, as previously mentioned, all messages are written to the WebSphere Application Server server's `SystemOut.log` file.

For each logger, the `jlog.properties` file defines an `isLogging` attribute which, when set to `true`, enables logging for that Tivoli Access Manager for WebSphere component. A value of `false` disables logging for that component.

`jlog.properties` defines "parent" loggers called MessageLogger and TraceLogger, which also have an `isLogging` attribute. If the "child" loggers do not specify this `isLogging` attribute they inherit the value of their respective parent. When Tivoli Access Manager for WebSphere is installed the `isLogging` attribute is set to `true` for the MessageLogger and `false` for the TraceLogger. This effectively means that message logging is enabled for all components and that tracing is disabled for all components.

To turn on tracing for a Tivoli Access Manager for WebSphere component two operations need to occur:

1. The `jlog.properties` file needs to be updated and the `isLogging` attribute set to `true` for the desired component. For example, to enable tracing for the AMWASWeb component the following line should be added to the `jlog.properties`:
   ```
   baseGroup.AMWASWebTraceLogger.isLogging=true
   ```

2. Enable tracing for the PDWAS component in the WebSphere Application Server console. To do this using the WebSphere Application Server console perform the following steps:
   **For WebSphere Version 5**
   a. Click Servers → Application Servers in the left frame.
   b. Click on the target server.
   c. Click Logging and Tracing → Diagnostic Trace.
   d. Under the Trace Specifications heading, click Modify.
   e. Click Components → PDWAS and select the required trace level.
   f. Click Apply.

   **For WebSphere version 4**
   a. Click Servers → Application Servers in the left frame.
b. Click on the target server.

c. Click Logging and Tracing Properties → Diagnostic Trace Service.

d. In the Trace Specification box, enter the following:
   com.ibm.ws.security.PDWAS=all=enabled

e. Click Apply.

The Trace Specification should now indicate that tracing is enabled at the required level. Save the configuration, and restart the server for the changes to take effect.

---

**Single Sign-On to WebSphere Application Server using WebSEAL**

Tivoli Access Manager WebSEAL can be used as a proxy server to provide access management and single sign-on capability to your Tivoli Access Manager for WebSphere protected applications. With such an architecture, WebSEAL authenticates users and forwards the collected credentials to the WebSphere Application Server in the form of an IV Header. WebSphere Trust Association Interceptors (TAIs) intercept requests from WebSEAL, extract the end-user name from the iv-user HTTP header and forward it to Tivoli Access Manager for WebSphere which uses the information to construct the client credential information and authorize the user. Refer to the WebSphere Application Server documentation for information about TAI.

Perform the following steps — detailed in the following sections — to set WebSEAL as an authentication proxy for WebSphere Application Server:

- “Step 1 — Create a trusted user account in Tivoli Access Manager”
- “Step 2 — Create a WebSEAL junction to the WebSphere Application Server”
- Either “Step 3a — Configure SSO using TAI for WebSphere Application Server version 4.0.6” on page 64 or “Step 3b — Configure SSO using TAI for WebSphere Application Server version 5.0.2” on page 64 depending on the WebSphere Application Server version you are using.

**Step 1 — Create a trusted user account in Tivoli Access Manager**

One of the underlying security requirements of the TAI is the creation of a trusted user account in the Tivoli Access Manager user registry that WebSphere Application Server is configured to use. This is the ID and password that WebSEAL uses to identify itself to the WebSphere Application Server. To prevent potential vulnerabilities, do not use sec_master as the trusted user account and ensure the password you use is unique. The trusted user account should be for the TAI only. On the Tivoli Access Manager machine, enter the commands below on the **padmin** command line:

```
padmin> user create webseal_userid webseal_userid_ON first_name surname password

padmin> user modify webseal_userid account-valid yes
```

**Step 2 — Create a WebSEAL junction to the WebSphere Application Server**

While WebSEAL can be configured to pass the end user identity in other ways, the iv-user header is the only one supported by the TAI. We recommend that communications over the junction use SSL for increased security. Setting up SSL across this junction requires that you configure the HTTP Server used by WebSphere Application Server, and WebSphere Application Server itself, to accept inbound SSL traffic and route it correctly to WebSphere Application Server. This
will include importing the necessary signing certificates into the WebSEAL certificate keystore, and possibly also the HTTP Server certificate keystore.

Create the WebSEAL junction to the WebSphere Application Server using the -c iv_user option. For example (entered as one line):

```bash
server task webseald-server create -t ssl -c iv_user -B
-U user -W password -h host_name junction_name -b supply
```

**Notes:**
1. If warning messages are displayed about the incorrect setup of certificates and key databases, delete the junction, correct problems with the key databases and re-create the junction.
2. The junction can be created as -t tcp or -t ssl depending on your requirements.

For more details and options about how to configure junctions between WebSEAL and WebSphere Application Server, including other options for specifying the WebSEAL server identity, refer to the WebSEAL Administration Guide as well as to the documentation for the HTTP Server you are using with your WebSphere Application Server.

**Step 3a — Configure SSO using TAI for WebSphere Application Server version 4.0.6**

To configure WebSEAL single sign-on for TAI, edit the TAI configuration file, $WAS_HOME/properties/webseal.properties, and ensure that the following parameters are set:

- `com.ibm.websphere.security.webseal.loginId` is set to the same username as created in step 1.
- The `hostnames` and `ports` parameters include the host name and port of the WebSEAL server.
- `com.ibm.websphere.security.webseal.id` is configured for the iv-user header. That is:

  ```
  com.ibm.websphere.security.webseal.id=iv-user
  ```

- Ensure that the following `trustedservers.properties` are correct:
  - `webseal` is listed in `com.ibm.websphere.security.trustassociation.types`
  - The webseal interceptor class has been set, that is (entered as one line):
    ```
    ```
  - The properties file is correct, that is:
    ```
    com.ibm.websphere.security.trustassociation.webseal.config=webseal
    ```

1. From the WebSphere Administration Console, access Authentication → Security Center and ensure Enable Web Trust Association is selected.
2. From the Virtual Hosts folder, select the default_host virtual host. Click Add in the Aliases box. Enter the new alias as `*:443`
3. Restart WebSphere.

**Step 3b — Configure SSO using TAI for WebSphere Application Server version 5.0.2**

The following steps are required when setting up security for the first time.

1. Click Security → Authentication mechanisms → LTPA in the left navigation panel.
2. Click **Trust Association** under **Additional Properties**.
3. Select the **Trust Association Enabled** check box.
4. Click **Interceptors under Additional Properties**.
5. Click *com.ibm.ws.security.web.WebSealTrustAssociationInterceptor* to use the WebSEAL interceptor. This interceptor is the default value.
6. Click **Custom Properties under Additional Properties**.
7. Click **New** to enter the property name and value pairs. Ensure the following parameters are set:
   - Ensure *webseal* is listed in *com.ibm.websphere.security.trustassociation.types*
   - *com.ibm.websphere.security.webseal.loginId* is set to the same username as created in step 1.
   - *com.ibm.websphere.security.webseal.id* is configured for the iv-user header. That is:
     *com.ibm.websphere.security.webseal.id=iv-user*
   - *com.ibm.websphere.security.webseal.hostnames* specifies the host names (case sensitive) that are expected in the request header. For example:
     *com.ibm.websphere.security.webseal.hostnames=host1*
   - *com.ibm.websphere.security.webseal.ports* specifies the corresponding port number of the host names that are expected in the request header. This should also include the proxy port numbers (if any) unless the *com.ibm.websphere.security.webseal.ignoreProxy* is set to true. For example:
     *com.ibm.websphere.security.webseal.ports=80,443*
   - *com.ibm.websphere.security.webseal.ignoreProxy* is an optional property that if set to *true* or *yes* ignores the proxy host names and ports in the IV header. By default this property is set to false.
8. Click OK. Save configuration and logout. Restart WebSphere Application Server.

**Step 4 — Set the SSO password in WebSEAL**

Edit the WebSEAL configuration file, *webseal_install_directory/etc/webseald-default.conf* and set the sso password for the user created in step 1 by setting the following parameter, *basicauth-dummy-passwd=webseal_userid_password*

Restart WebSEAL.

**Step 5 — Test the WebSEAL connection**

To ensure access to WebSphere Application Server across the WebSEAL junction is operating correctly, log onto WebSEAL once and attempt to access protected objects on the WebSphere Application Server across the junction.

**Troubleshooting tips**

The section contains the following topics:

- “WebSphere server does not start after configuration and migration — WebSphere Application Server version 4.0.6 only” on page 66
- “WebSphere server does not start after unconfiguration — WebSphere Application Server version 4.0.6 only” on page 66
WebSphere server does not start after configuration and migration — WebSphere Application Server version 4.0.6 only

Problem: After configuring Tivoli Access Manager for WebSphere, the WebSphere Application Server will not start.

Explanation: There are two possible reasons:

- During Tivoli Access Manager for WebSphere configuration, the new administrative group pdwas-admin was not added to the appropriate ACLs.
- During Tivoli Access Manager for WebSphere configuration, the new administrative group pdwas-admin was added to the appropriate ACLs, but the ACLs were not updated to all authorization servers. This problem can only occur in secure domains that have more than one authorization server.

Solution:

There are two possible solutions:

- If pdwas-admin was not added to the appropriate ACLs, add it now. See the instructions for adding the pdwas-admin group to the administration ACL in "Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6" on page 31.
- If pdwas-admin was added to the appropriate ACLs, and the secure domain contains more than one authorization server, and the authorization servers were not updated, update them now. See the instructions for adding the pdwas-admin group to the administration ACL in "Part 5a: Migrate WebSphere security settings — WebSphere version 4.0.6" on page 31.

WebSphere server does not start after unconfiguration — WebSphere Application Server version 4.0.6 only

Problem: After unconfiguring Tivoli Access Manager for WebSphere and the Access Manager Java Runtime Environment, the WebSphere Application Server might not start. This problem occurs very intermittently. WebSphere Application Server fails to load the security collaborator, com.ibm.ejs.security.EJSSecurityCollaborator.

Workaround: Disable WebSphere Application Server security and restart WebSphere Application Server.

1. Go to the system running DB2. Log on as the user name with which DB2 was installed. For example:

   ```
   # su - db2inst1
   ```

   A usage message is displayed

2. Enter the following commands as shown in bold font, where was40 is the name of the WebSphere version 4 database:

   ```
   db2 => connect to was40 user db2inst1
   ```

   Enter current password for db2inst1:

   Database Connection Information
   Database Server = DB2/LINUX 7.2.0
   SQL authorization ID = DB2INST1
   Local database alias = WAS40

   ```
   db2 => update ejssadmin.securitycfg_table set securityenabled = 0
   DB20000I The SQL command completed successfully
   ```
3. Start WebSphere Application Server.

**Backing up Tivoli Access Manager for WebSphere files**

It is good administrative practice to employ a backup strategy for Tivoli Access Manager for WebSphere files so that vital information can be restored in the event of a critical failure.

The key Tivoli Access Manager for WebSphere files are:

- PDWAS.properties and jlog.properties located in the /etc directory of the Tivoli Access Manager for WebSphere installation.
- PD_WAS.prop located in the /config directory of the WebSphere Application Server installation.

To restore Tivoli Access Manager for WebSphere, the application should be reinstalled and the above files copied back to the appropriate locations on the Tivoli Access Manager for WebSphere and WebSphere Application Server installations.
Chapter 6. Tutorial: How to enable security

This chapter provides a tutorial that describes how to add security to an example application. The tutorial is based on a WebSphere tutorial that helps you learn about various aspects of WebSphere application assembly, configuration, and deployment. The WebSphere tutorial accompanies example code that is included as part of the WebSphere product.

You do not need to consult the WebSphere tutorial to use this Tivoli Access Manager tutorial. This Tivoli Access Manager for WebSphere tutorial provides an application EAR file that has been built from the WebSphere example code by following the WebSphere tutorial instructions.

WebSphere tutorials can be found online at:

The example program that is included with Tivoli Access Manager for WebSphere is built from the tutorial instructions in Sections 6.7.1, 6.7.2, and 6.7.3 at the Web site listed above. The material in this chapter replaces the tutorial in Section 6.7.4 at the Web site listed above.

This chapter includes the following sections:

- “Tutorial : for Tivoli Access Manager for WebSphere Application Server version 4.0.6” on page 78
- “Tutorial : for Tivoli Access Manager for WebSphere Application Server version 5.0.2” on page 78

Tutorial : for Tivoli Access Manager for WebSphere Application Server version 4.0.6

How to use the tutorial

This tutorial shows you how to add security to the application EAR file, add users to the LDAP user registry, enable WebSphere security, deploy and test the sample application, migrate the application to Tivoli Access Manager, enable the Tivoli Access Manager for WebSphere authorization component, and test the application security under Tivoli Access Manager. The tutorial also shows you how to make a simple change to a role, and then test that the result is recognized during access checking.

These instructions assume the following:

- WebSphere Application Server has been installed and configured to use an IBM Directory LDAP Server.
- Security has not been enabled for WebSphere.

You can run this tutorial either before or after completing the initial installation and configuration of Tivoli Access Manager for WebSphere. If you have not yet installed Tivoli Access Manager for WebSphere, the tutorial will instruct you when to install it.
These instructions assume that Tivoli Access Manager and WebSphere Application Server are installed and configured, and they are using the same IBM Directory Server user registry.

If you have not yet installed and configured Tivoli Access Manager for WebSphere, complete the instructions in each of the following sections:

- **Part 1: Add users to the LDAP user registry**
- **Part 2: Install Tivoli Access Manager for WebSphere** on page 71
- **Part 3: Add security to a WebSphere application** on page 71
- **Part 4: Create the Tivoli Access Manager administrative user for WebSphere Application Server** on page 73
- **Part 5: Enable WebSphere security** on page 73
- **Part 6: Deploy the application** on page 74
- **Part 7: Test security for the deployed application** on page 75
- **Part 8: Migrate the application to Tivoli Access Manager** on page 75
- **Part 9: Test security for the deployed application** on page 77
- **Part 10: Change roles** on page 77
- **Part 11: Test security for the deployed application** on page 78

If you have completed configuration the initial installation of Tivoli Access Manager for WebSphere, according to the instructions in Chapter 3, “Configuration procedures,” on page 25, you need to complete only the following sections:

- **Part 1: Add users to the LDAP user registry**

**Note:** You do not need to perform Step 2 in this part. This task was completed during initial configuration of Tivoli Access Manager for WebSphere.

- **Part 3: Add security to a WebSphere application** on page 71
- **Part 6: Deploy the application** on page 74
- **Part 7: Test security for the deployed application** on page 75
- **Part 8: Migrate the application to Tivoli Access Manager** on page 75
- **Part 9: Test security for the deployed application** on page 77
- **Part 10: Change roles** on page 77
- **Part 11: Test security for the deployed application** on page 78

**Part 1: Add users to the LDAP user registry**

Use the Tivoli Access Manager pdadmin utility to add the users you declared in the previous section (user1, user2, and user3) to the LDAP user registry. Also add an additional user, user4.

This section demonstrates common pdadmin commands for adding users. For complete information on all pdadmin options, see the IBM Tivoli Access Manager Base Administration Guide.

1. Log in as the Tivoli Access Manager administrator:
   ```c:
padmin -a sec_master -p myPassword
   ```

   Substitute the correct password for the sec_master account for your Tivoli Access Manager secure domain.

2. If you have already installed Tivoli Access Manager for WebSphere, and completed the initial configuration, skip this step. Go to the next step.
If you have not yet installed Tivoli Access Manager for WebSphere, create a WebSphere administration user. Enter the following command as one continuous command line:

```
pdadmin> user create wsadmin cn=wsadmin,o=organization,c=country wsadmin wsadmin myPassword
```

Substitute values for organization and country that are valid for your LDAP user registry.

3. Create user accounts for each of the new users. Assign passwords. The following examples show sample commands, where the organization is ibm and the country is au, and all users receive the password myPassword.

```
pdadmin> user create user1 cn=user1,o=ibm,c=us user1 myPassword
pdadmin> user create user2 cn=user2,o=ibm,c=us user2 myPassword
pdadmin> user create user3 cn=user3,o=ibm,c=us user3 myPassword
pdadmin> user create user4 cn=user4,o=ibm,c=us user4 myPassword
```

4. Enable all the accounts:

```
pdadmin> user modify wsadmin account-valid yes
pdadmin> user modify user1 account-valid yes
pdadmin> user modify user2 account-valid yes
pdadmin> user modify user3 account-valid yes
pdadmin> user modify user4 account-valid yes
```

5. Exit the pdadmin utility:

```
pdadmin> quit
```

6. Return to the WebSphere console to enable security. Continue to "Part 5: Enable WebSphere security" on page 73.

**Part 2: Install Tivoli Access Manager for WebSphere**

If you have already installed and configured Tivoli Access Manager for WebSphere, skip this part. Go to the next part "Part 8: Migrate the application to Tivoli Access Manager" on page 75.

You are now ready to install and configure the Tivoli Access Manager for WebSphere software.

Follow the instructions in Chapter 2, “Installation instructions,” on page 11.

After installing the Tivoli Access Manager for WebSphere files, complete the initial configuration described in “Configuring the initial installation” on page 25, with the following exception:

You have already created the WebSphere administrative user, wsadmin, during this tutorial, in "Part 1: Add users to the LDAP user registry” on page 70. Thus you do not need to do this during the initial configuration. Therefore, skip step 2 in "Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26.

**Part 3: Add security to a WebSphere application**

1. Start the WebSphere application assembly tool. Click Start → Programs → IBM WebSphere → Application Server v4.0 AE → Application Assembly Tool or run C:\WebSphere\AppServer\bin\assembly

Click Cancel at the Welcome screen.
2. Copy the sample application file simpleSession.ear from the directory where it was extracted to C:\temp\assembly\simpleSession.ear

3. From the WebSphere application assembly tool, open the sample application EAR file. Click File → Open C:\temp\assembly\simpleSession.ear


5. Select the General tab. Add:
   
   Name: GoodGuys

6. Select the Bindings tab. Click Add user.
   
   Name: user1

   Click OK.

7. Repeat the previous step to add the following users:
   
   Name: user2
   Name: user3

   Click OK when all users are added.

   
   Name: MyMethodPermissions

   a. Method: Click Add.
      
      • Select Home (*)
      • Select Remote (*)

      Click OK.

   b. Roles: Click Add. Select GoodGuys. Click OK.

   
   a. Click the Advanced tab.
   
   b. Check the Login Configuration box.
   
   c. Specify the Authorization Method: Basic.
   
   d. Specify the Realm Name: Getting Started
   
   e. Click Apply.

    
    a. For Security Constraint Name, enter GoodGuys.
    
    b. Roles:
       
       • Click Add.
       
       • Select GoodGuys.
       
       • Click OK.
    
    c. For Transport Guarantee, select None.
    
    d. Click OK.

    
    a. Select New.
    
    b. For Web Resource Name, enter SecureMe.
    
    c. For HTTP Methods, click Add. Select GET. Click OK.
    
    d. For HTTP Methods, click Add. Select POST. Click OK.
    
    e. For URLS, click Add. Enter: "/SimpleSession". Click OK.
    
    f. Click OK.
12. Save the new EAR file. Select File->Save As and enter:
   C:\temp\assembly\simpleSessionSecure.ear
    a. Set working directory to C:\temp.
    b. Click Generate Now.
    c. Fix any errors.
14. Exit the Application Assembly tool. Continue to the next section “Part 1: Add users to the LDAP user registry” on page 70.

Part 4: Create the Tivoli Access Manager administrative user for WebSphere Application Server

If security has already been enabled in WebSphere Application Server the WebSphere Application Server administrative user should be imported into the Tivoli Access Manager object space. Use either the Tivoli Access Manager command line utility, pdadmin, or the Tivoli Access Manager Web Portal Manager to import the Tivoli Access Manager administrative user for WebSphere Application Server. To do this from the Tivoli Access Manager command line utility:

1. From a command line, start pdadmin as administrative user sec_master:
   pdadmin -a sec_master -p sec_master_password
2. Import the WebSphere Application Server administrative user. For example:
   pdadmin> user import was_admin_user dn_registry_identifier

   Make the WebSphere administrative user account valid:
   pdadmin> user modify was_admin_user account-valid yes

If security has not been enabled in WebSphere Application Server the WebSphere Application Server administrative user needs to be created. Use either the Tivoli Access Manager command line utility, pdadmin, or the Tivoli Access Manager Web Portal Manager to create the Tivoli Access Manager administrative user for WebSphere Application Server.

The following instructions describe how to use pdadmin.

1. From a command line, start pdadmin as administrative user sec_master:
   pdadmin -a sec_master -p sec_master_password
2. Create a Tivoli Access Manager administrative user for WebSphere Application Server. For example, the following instructions create a new user wsadmin. The following command must be entered as one continuous command line:
   pdadmin> user create wsadmin cn=wsadmin,o=organization,c=country
   wsadmin myPassword

   Substitute values for organization and country that are valid for your LDAP user registry.

   Make the wsadmin account valid:
   pdadmin> user modify wsadmin account-valid yes

Part 5: Enable WebSphere security

If you have not enabled WebSphere security then do so using the instructions in “Enable security in WebSphere Application Server version 4.0.6” on page 27.
Part 6: Deploy the application

1. Verify that the WebSphere Administration Server is running.
2. Start the WebSphere Administration Client:
   C:\websphere\appserver\bin\adminclient
3. Log in as user wsadmin with password myPassword.
4. Select WebSphere Admin Domain -> Enterprise Applications.
5. Right click and select Install Enterprise Application.
   a. Check Install Application button.
   b. Set the path:
      c:\temp\assembly\simpleSessionSecure.ear
   c. Click Next. A dialog box prompts you to deny access to all unprotected methods. Click yes.
   d. Click Select.
   e. Verify that all users are listed
      user1 user2 user3
   f. Click OK.
   g. You can now select Next at each of a series of dialog boxes that open. The dialog boxes are titled:
      • Mapping Users to Roles
      • Mapping EJB RunAs Role to User
      • Binding Enterprise Bean to JNDI Names
      • Mapping EJB References to Resources
      • Specifying the Default Datasources for EJB Modules
      • Specifying Data Sources for Individual CMP Beans
      • Selecting Virtual Hosts for Web Modules
      • Selecting Application Server
   h. When the dialog box Completing the Application Installation Wizard opens, click Finish.
   i. Click Yes to generate code. Click OK.
   j. Click OK to exit the dialog box.
6. If the default server is running, stop it now. If the default server is not running, continue to the next step.
   To stop the default server:
      • Select WebSphere Admin Domain -> Nodes -> hostname -> Application Servers -> Default Server
      • Right click on the default server.
      • Select Stop.
      • Click OK to exit the dialog box.
7. Start the Default Server.
   • Select WebSphere Admin Domain -> Nodes -> hostname -> Application Servers -> Default Server
   • Right click on the default server.
   • Select Start.
   • Click OK to exit the dialog box.
8. Exit the WebSphere Advanced Administration Console.
9. Continue to “Part 7: Test security for the deployed application” on page 75.
Part 7: Test security for the deployed application

Servlet
1. Start your Web browser.
2. Go to the following URL. Substitute your system name for hostname:
3. You should be prompted to enter a user name and password. Enter one of the valid user names: user1 or user2 or user3, and enter one invalid name, such as user4. Enter the correct password.
   You should see a results page. When you enter the invalid name user4, you should see a failure page.
4. Restart your Web browser.
5. Go to the same URL. When prompted to enter a user name and password, enter an invalid user name or password.
   You should see a failure page.

Thick Client
1. Use the launchclient program to start your secure application. Enter the following command as one single line:
   C:> c:\websphere\appserver\bin\launchclient
   c:\websphere\appserver\installedApps\simpleSessionSecure.ear
2. You should receive a login prompt, requesting a user name and password.
3. Enter a valid user name and password. For example, user1.
   You should see text indicating a successful login.
4. Restart your Web browser.
5. Use the launchclient program to start your secure application, as shown in Step 1 above. When prompted to enter a user name and password, enter an invalid user name or password.
   You should see text indicating a login failure.
6. Continue to the next section.

Part 8: Migrate the application to Tivoli Access Manager

These instructions assume that you have completed the initial installation and configuration of Tivoli Access Manager for WebSphere, as described in “Configuring the initial installation” on page 25. The initial installation and configuration included the migration of the admin.ear file.

Note: If you have not completed an initial installation and configuration of Tivoli Access Manager for WebSphere, complete it now. See the instructions in “Configuring the initial installation” on page 25.

1. Assemble the following information, which you will need to specify as input parameters to the migration utility:
   • The name of the EAR file to migrate:
     c:\temp\assembly\simpleSessionSecure.ear
   • The location of the PDPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.

Note: The file location must be expressed as a Uniform Resource Indicator.
   – Solaris, Linux, HP-UX
The name of the Tivoli Access Manager administration account. This should be sec_master.

The password for the sec_master account.

The name of the WebSphere administrative user account. This should match the account you created during the initial configuration of Tivoli Access Manager for WebSphere. For example:

wsadmin

The LDAP distinguished name (DN) suffix under which both the Tivoli Access Manager policy server and WebSphere Application Server store user information. This should match the DN suffix used when you created the wsadmin user.

The example shown in "Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server" on page 26 created wsadmin with following DN:

cn=wsadmin,o=ibm,c=us

In this case the DN suffix is: o=ibm,c=us

This value should be given as the argument to the –d option to the migrateEAR4 utility.

Note: You can use pdadmin to display the DN for wsadmin on your system:

padmin> user show wsadmin

2. Change directory to the location of the migration utility:

• (UNIX) /opt/pdas/bin
• (Windows) C:\Program Files\Tivoli\pdas\bin
3. Run the migration utility to migrate the application data.
   Using the parameters that you assembled in the previous step, enter the following text at a command prompt, as one continuous command line:

   **Table 8. Command line invocation of the migration utility**

<table>
<thead>
<tr>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrateEAR4 -j /temp/assembly/simpleSessionSecure.ear</td>
</tr>
<tr>
<td>-a sec_master -p sec_master_password</td>
</tr>
<tr>
<td>-w wsadmin -d &quot;o=ibm,c=us&quot;</td>
</tr>
<tr>
<td>-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Note that the default location of the PdPerm.properties file on AIX is:</td>
</tr>
<tr>
<td>/usr/WebSphere/AppServer/java/jre/PdPerm.properties</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>migrateEAR4 -j C:\temp\assembly\simpleSessionSecure.ear</td>
</tr>
<tr>
<td>-a sec_master -p sec_master_password</td>
</tr>
<tr>
<td>-w wsadmin -d &quot;o=ibm,c=us&quot;</td>
</tr>
<tr>
<td>-c file:C:\WebSphere\AppServer\java\jre\PdPerm.properties</td>
</tr>
</tbody>
</table>

   The migration utility logs output to a log file. The name of the log file is displayed. For example, pdwas_migrate.log. You can examine the contents of the log file to verify that all roles were migrated.
   If the log file does not appear, the migration utility encountered a problem. If this occurs, verify that you supplied the correct Uniform Resource Indicator to the -c option, and the correct filename to the -j option.

4. When the script completes, continue to the next section “Part 9: Test security for the deployed application.”

   **Part 9: Test security for the deployed application**

   1. Verify that security is now working for the application. Repeat the steps for both Servlet and Thick Client in “Part 7: Test security for the deployed application” on page 75.
   2. When security has been verified, continue to “Part 10: Change roles.”

   **Part 10: Change roles**

   Use the Tivoli Access Manager **pdadmin** utility to change a role definition by adding a user.

   1. Start **pdadmin**:
      pdadmin -a sec_master -p myPassword
   2. Modify the ACL for the SimpleSession application to add the name of user4.
      Enter the acl modify command below as one continuous command line:
      pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_SimpleSessApp_ACL
      set user user4 T[WebAppServer]
   3. Replicate to the server and exit the utility:
      pdadmin> server replicate
      pdadmin> quit
   4. Continue to “Part 11: Test security for the deployed application” on page 78.
Part 11: Test security for the deployed application

1. Verify that security is now working for the application. Repeat the steps for both Servlet and Thick Client in “Part 7: Test security for the deployed application” on page 75.
   Note that when you enter the name of a valid user, you can now enter either user1, user2, user3 or user4.

2. Verify that user4 is now able to log in.

You have now completed the tutorial.

---

Tutorial: for Tivoli Access Manager for WebSphere Application Server version 5.0.2

How to use the tutorial

This tutorial shows you how to add security to the application EAR file, add users to the LDAP user registry, enable WebSphere security, deploy and test the sample application, migrate the application to Tivoli Access Manager, enable the Tivoli Access Manager for WebSphere authorization component, and test the application security under Tivoli Access Manager. The tutorial also shows how to make a simple change to a role, and then test that the result is recognized during access checking.

These instructions assume the following:

- WebSphere Application Server has been installed and configured to use an IBM Directory Server.
- Security has not been enabled for WebSphere.

You can run this tutorial either before or after completing the initial installation and configuration of Tivoli Access Manager for WebSphere. If you have not yet installed Tivoli Access Manager for WebSphere, the tutorial will instruct you when to install it.

These instructions assume that Tivoli Access Manager and WebSphere Application Server are installed and configured, and they are using the same IBM Directory Server user registry.

If you have not yet installed and configured Tivoli Access Manager for WebSphere, complete the instructions in each of the following sections:

- “Part 1: Add users to the LDAP user registry” on page 79
- “Part 2: Install Tivoli Access Manager for WebSphere” on page 80
- “Part 3: Add security to a WebSphere application” on page 80
- “Part 4: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 82
- “Part 5: Enable WebSphere security” on page 82
- “Part 6: Deploy the application” on page 82
- “Part 7: Test security for the deployed application” on page 83
- “Part 8: Migrate the application to Tivoli Access Manager” on page 84
- “Part 9: Test security for the deployed application” on page 85
- “Part 10: Change roles” on page 85
- “Part 11: Test security for the deployed application” on page 86
If you have completed configuration the initial installation of Tivoli Access Manager for WebSphere, according to the instructions in Chapter 3, “Configuration procedures,” on page 25, you need to complete only the following sections:

- “Part 1: Add users to the LDAP user registry”
- “Part 3: Add security to a WebSphere application” on page 80

Note: You do not need to perform Step 3 in this part. This task was completed during initial configuration of Tivoli Access Manager for WebSphere.

- “Part 6: Deploy the application” on page 82
- “Part 7: Test security for the deployed application” on page 83
- “Part 8: Migrate the application to Tivoli Access Manager” on page 84
- “Part 9: Test security for the deployed application” on page 85
- “Part 10: Change roles” on page 85
- “Part 11: Test security for the deployed application” on page 86

**Part 1: Add users to the LDAP user registry**

Use the Tivoli Access Manager **pdadmin** utility to add the users you declared in the previous section (user1, user2, and user3) to the LDAP user registry. Also add an additional user, user4.

This section demonstrates common **pdadmin** commands for adding users. For complete information on all **pdadmin** options, see the IBM Tivoli Access Manager Base Administration Guide.

1. Log in as the Tivoli Access Manager administrator:
   
   C:> pdadmin -a sec_master -p myPassword

   Substitute the correct password for the sec_master account for your Tivoli Access Manager secure domain.

2. If you have already installed Tivoli Access Manager for WebSphere, and completed the initial configuration, skip this step. Go to the next step.

   If you have not yet installed Tivoli Access Manager for WebSphere, create a WebSphere administration user. Enter the following command as one continuous command line:

   ```
   pdadmin> user create wsadmin cn=wsadmin,o=organization,c=country wsadmin wsadmin myPassword
   ```

   Substitute values for organization and country that are valid for your LDAP user registry.

3. Create user accounts for each of the new users. Assign passwords. The following examples show sample commands, where the organization is ibm and the country is us, and all users receive the password myPassword.

   ```
   pdadmin> user create user1 cn=user1,o=ibm,c=us user1 user1 myPassword
   pdadmin> user create user2 cn=user2,o=ibm,c=us user2 user2 myPassword
   pdadmin> user create user3 cn=user3,o=ibm,c=us user3 user3 myPassword
   pdadmin> user create user4 cn=user4,o=ibm,c=us user4 user4 myPassword
   ```

4. Enable all the accounts:
pdadmin> user modify wsadmin account-valid yes
pdadmin> user modify user1 account-valid yes
pdadmin> user modify user2 account-valid yes
pdadmin> user modify user3 account-valid yes
pdadmin> user modify user4 account-valid yes

5. Exit the pdadmin utility:
padmin> quit

6. Return to the WebSphere console to enable security. Continue to “Part 5: Enable WebSphere security” on page 82.

Part 2: Install Tivoli Access Manager for WebSphere

If you have already installed and configured Tivoli Access Manager for WebSphere, skip this part. Go to the next part, “Part 8: Migrate the application to Tivoli Access Manager” on page 84.

You are now ready to install and configure the Tivoli Access Manager for WebSphere software.

Follow the instructions in Chapter 2, “Installation instructions,” on page 11.

After installing the Tivoli Access Manager for WebSphere files, complete the initial configuration described in “Configuring the initial installation” on page 25, with the following exception:

You have already created the WebSphere administrative user, wsadmin, during this tutorial, in “Part 1: Add users to the LDAP user registry” on page 79. Thus you do not need to do this during the initial configuration. Therefore, skip step 2 in “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26.

Part 3: Add security to a WebSphere application

1. Copy the sample application file simpleSession.ear from the %PDWAS_HOME%\example directory (where it was extracted) to C:\temp\assembly\simpleSession.ear

2. Start the WebSphere application assembly tool. Click Start->Programs->IBM WebSphere -> Application Server v5.0 -> Application Assembly Tool or run C:\WebSphere\AppServer\bin\assembly
   Click Cancel at the Welcome screen.

3. From the WebSphere application assembly tool, open the sample application EAR file. Click File -> Open C:\temp\assembly\simpleSession.ear

   Name: GoodGuys
   Click OK.

5. Expand Web Modules. Expand SimpleSessionWar. Right click Security Roles. Click New. Add:
   Name: GoodGuys
   Click OK.

6. From the top level, right click Security Roles. Select the Bindings tab. From the Users section, click Add. Enter user1.
   Click OK.
7. Repeat the previous step to add the following users:
   Name: user2
   Name: user3

   Click Apply when all users are added.

8. Expand EJB Modules. Expand EJB11. Right click Method Permissions. Select New. Click on the Add button and enter into the Method Permission_Name: field:
   Name: 'MyMethodPermissions'
   a. From the Method section, click Add. Expand SimpleSessionEJB10.jar then com_ibm_websphere_gettingstarted_ejbs_SimpleSession_(*). Select All Methods. Click OK.
   b. From the Roles section, click Add. Select GoodGuys. Click OK.
   c. Click OK.

   a. Click the Advanced tab.
   b. Check the Login Configuration box.
   c. Specify the Authorization Method: Basic.
   d. Specify the Realm Name: Getting Started
   e. Click Apply.


   a. For Security Constraint Name, enter GoodGuys.
   b. Roles:
      • Click Add.
      • Select GoodGuys.
   c. For Transport Guarantee, select None.
   d. Click OK.

   a. Right-click and select New.
   b. For Web Resource Name, enter SecureMe.
   c. For HTTP Methods, click Add. Select GET. Click OK.
   d. For HTTP Methods, click Add. Select POST. Click OK.
   e. For URLs, click Add. Enter: "/SimpleSession". Click OK.
   f. Click OK.

13. Save the new EAR file. Select File->Save As and enter:
    C:\temp\assembly\simpleSessionSecure.ear

    a. Set working directory to C:\temp.
    b. Click on the Generate Now button.
    c. Fix any errors.
    d. Click on the Close button.

15. Exit the Application Assembly tool. Continue to the next section [Part 1: Add users to the LDAP user registry] on page 70.
Part 4: Create the Tivoli Access Manager administrative user for WebSphere Application Server

If security has already been enabled in WebSphere Application Server the WebSphere Application Server administrative user should be imported into the Tivoli Access Manager object space. Use either the Tivoli Access Manager command line utility, *pdadmin*, or the Tivoli Access Manager Web Portal Manager to import the Tivoli Access Manager administrative user for WebSphere Application Server. To do this from the Tivoli Access Manager command line utility:

1. From a command line, start *pdadmin* as administrative user sec_master:
   
   ```
   pdadmin -a sec_master -p sec_master_password
   ```

2. Import the WebSphere Application Server administrative user. For example:
   
   ```
   pdadmin> user import was_admin_user dn_registry_identifier
   ```

   Make the WebSphere administrative user account valid:
   
   ```
   pdadmin> user modify was_admin_user account-valid yes
   ```

If security has not been enabled in WebSphere Application Server the WebSphere Application Server administrative user needs to be created. Use either the Tivoli Access Manager command line utility, *pdadmin*, or the Tivoli Access Manager Web Portal Manager to create the Tivoli Access Manager administrative user for WebSphere Application Server.

The following instructions describe how to use *pdadmin*.

1. From a command line, start *pdadmin* as administrative user sec_master:
   
   ```
   pdadmin -a sec_master -p sec_master_password
   ```

2. Create a Tivoli Access Manager administrative user for WebSphere Application Server. For example, the following instructions create a new user *wsadmin*. The following command must be entered as one continuous command line:
   
   ```
   pdadmin> user create wsadmin cn=wsadmin,o=organization,c=country
   wsadmin wsadmin myPassword
   ```

   Substitute values for organization and country that are valid for your LDAP user registry.

   Make the wsadmin account valid:
   
   ```
   pdadmin> user modify wsadmin account-valid yes
   ```

Part 5: Enable WebSphere security

If you have not enabled WebSphere security then do so using the instructions in "Enable security in WebSphere Application Server version 5.0.2" on page 27.

Part 6: Deploy the application

1. Ensure that the WebSphere Administration Server is running.
2. Open the Administration Console: http://localhost:9090/admin.

   **Note:** Once LTPA security is enabled, you must use the FQDN:
   
   http://hostname.domain.com:9090/admin

3. Logon as wsadmin.
4. Select **Enterprise Applications** then **Install New Application**
5. Click **Browse** to locate the application, that is, C:\temp\assembly\simpleSessionSecure.ear. Click **Open**.

6. You can now select **Next** on a series of screens that are displayed. The screens are titled:
   - Preparing for the application installation,
   - Step One: Provide options to perform the installation,
   - Step Two: Provide options to perform the EJB Deploy,
   - Step Three: Provide JNDI Names for Beans,
   - Step Four: Map EJB references to beans,
   - Step Five: Map virtual hosts for web modules,
   - Step Six: Map modules to application servers,
   - Step Seven: Map security roles to users/groups,
   - Step Eight: Correct use of System Identity,
   - Step Nine: Summary.

   Click **Finish** to begin the installation of the application.

7. Click on the **Save to Master Configuration** link.

8. Click on the **Save** button to confirm the **Save to Master Configuration**.

9. Start the application by selecting **Enterprise Applications**, locating **SimpleSessionApp**, ticking the checkbox and selecting the **Start** button.

10. Click **Start**.

### Part 7: Test security for the deployed application

#### Servlet

1. Start your Web browser.

2. Go to the following URL. Substitute your system name for hostname:
   

3. You should be prompted to enter a user name and password. Enter one of the valid user names: user1 or user2 or user3, and enter one valid user name which does not have permissions to the ACL, such as user4. Enter the correct password.

   You should see a results page containing the text "Test". When you enter the invalid name user4, you should see an unauthorized (403 Forbidden) page.

4. Restart your Web browser.

5. Go to the same URL. At the prompt, enter a nonexistent user name and password.

   You should be prompted to log in again.

#### Thick Client

1. Use the `launchclient` program to start your secure application. Enter the following command as one single line:
   
   C:> c:\program files\websphere\appserver\bin\launchclient
   "c:\program files\websphere\appserver\installedApps\<nodename>\simpleSessionSecure.ear"

2. You should receive a login prompt, requesting a user name and password.

3. Enter a valid user name and password. For example, user1.

   You should see text indicating a successful login.

4. Restart your Web browser.
5. Use the launchclient program to start your secure application, as shown in Step 1 above. When prompted to enter a user name and password, enter an invalid user name or password. You should see text indicating a login failure.

6. Continue to the next section.

Part 8: Migrate the application to Tivoli Access Manager

These instructions assume that you have completed the initial installation and configuration of Tivoli Access Manager for WebSphere, as described in “Configuring the initial installation” on page 25. The initial installation and configuration included the migration of the adminconsole.ear file.

Note: If you have not completed an initial installation and configuration of Tivoli Access Manager for WebSphere, complete it now. See the instructions in “Configuring the initial installation” on page 25.

1. Assemble the following information, which you will need to specify as input parameters to the migration utility:

   - The name of the EAR file to migrate:
     c:\temp\assembly\simpleSessionSecure.ear

   - The full path of the PdPerm.properties file. This file is located in a directory under the WebSphere Application Server installation directory. The following list shows the default location on each operating system.

     Note: The file location must be expressed as a Uniform Resource Indicator.
     - Solaris, Linux, HP-UX
       file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
     - AIX
       file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
     - Windows
       file:/'c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

   - The name of the Tivoli Access Manager administration account. This should be sec_master.
   - The password for the sec_master account.
   - The name of the WebSphere administrative user account. This should match the account you created during the initial configuration of Tivoli Access Manager for WebSphere. For example:
     wsadmin

   - The LDAP distinguished name (DN) suffix under which both the Tivoli Access Manager policy server and WebSphere Application Server store user information. This should match the DN suffix used when you created the wsadmin user.

     The example shown in “Part 1: Create the Tivoli Access Manager administrative user for WebSphere Application Server” on page 26 created wsadmin with following DN:
     cn=wsadmin,o=ibm,c=us

     In this case the DN suffix is: o=ibm,c=us

     This value should be given as the argument to the –d option to the migrateEAR5 utility.
Note: You can use `pdadmin` to display the DN for wsadmin on your system:

```
pdadmin> user show wsadmin
```

2. Change directory to the location of the migration utility:
   - (UNIX) /opt/amwas/bin
   - (Windows) C:\Program Files\Tivoli\amwas\bin

3. Run the migration utility to migrate the application data.
   Using the parameters that you assembled in the previous step, enter the following text at a command prompt, as one continuous command line:

```
Table 9. Command line invocation of the migration utility

<table>
<thead>
<tr>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrateEAR5 -j /temp/assembly/simpleSessionSecure.ear</td>
</tr>
<tr>
<td>-a sec_master -p sec_master_password</td>
</tr>
<tr>
<td>-w wsadmin -d &quot;o=ibm,c=us&quot;</td>
</tr>
<tr>
<td>-c file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties</td>
</tr>
</tbody>
</table>

Note that the default location of the PdPerm.properties file on AIX is:
/usr/WebSphere/AppServer/java/jre/PdPerm.properties

<table>
<thead>
<tr>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>migrateEAR5.bat -j C:\temp\assembly\simpleSessionSecure.ear</td>
</tr>
<tr>
<td>-a sec_master -p sec_master_password</td>
</tr>
<tr>
<td>-w wsadmin -d &quot;o=ibm,c=us&quot;</td>
</tr>
</tbody>
</table>
| -c file:"C:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

The migration utility logs output to a log file. The name of the log file is displayed. For example, `pdwas_migrate.log`. You can examine the contents of the log file to verify that all roles were migrated.

If the log file does not appear, the migration utility encountered a problem. If this occurs, verify that you supplied the correct Uniform Resource Indicator to the `-c` option, and the correct filename to the `-j` option.

4. When the script completes, continue to the next section "Part 9: Test security for the deployed application."

Part 9: Test security for the deployed application

1. Verify that security is now working for the application. Repeat the steps for both Servlet and Thick Client in "Part 7: Test security for the deployed application" on page 83.

2. When security has been verified, continue to "Part 10: Change roles."

Part 10: Change roles

Use the Tivoli Access Manager `pdadmin` utility to change a role definition by adding a user.

1. Start `pdadmin`:
   
   `pdadmin -a sec_master -p myPassword`

2. Modify the ACL for the SimpleSession application to add the name of user4. Enter the acl modify command below as one continuous command line:
   
   `pdadmin> acl modify _WebAppServer_deployedResources_GoodGuys_SimpleSessApp_ACL set user user4 T[WebAppServer]`

3. Replicate to the server and exit the utility:
   
   `pdadmin> server replicate`
   `pdadmin> quit`
4. Continue to “Part 11: Test security for the deployed application.”

**Part 11: Test security for the deployed application**

1. Verify that security is now working for the application. Repeat the steps for both Servlet and Thick Client in “Part 7: Test security for the deployed application” on page 83.

   Note that when you enter the name of a valid user, you can now enter either user1, user2, user3 or user4.

2. If you are unable to access the page as user4, wait for the cache to timeout or restart the WebSphere Application Server.

You have now completed the tutorial.
Chapter 7. Removal instructions

Remove Tivoli Access Manager for WebSphere by changing the necessary configuration files and removing the authorization component.

Begin by unconfiguring the Tivoli Access Manager for WebSphere authorization component:

1. Log in as:
   - (UNIX) root
   - (Windows) Windows’s user with administrator privileges.
2. Stop the WebSphere Application Server.
3. Run the **pdwascfg** utility to unconfigure the Tivoli Access Manager for WebSphere authorization component using the **-action unconfigWAS4** or **unconfigWAS5** parameter depending on the version of WebSphere Application Server that is running:

   ```
   # pdwascfg -action unconfigversion_number
   -remote_acl_user user_CN
   -sec_master_pwd password
   -was_home home_directory_of_WebSphere_Application_Server
   -pdmgrd_host policy_server_host_name
   -pdacld_host authorization_server_host_name
   ```

   For more information on the command line options, see the reference page for the command in [Appendix A, “Command reference,” on page 91](#).

To complete the removal, go to the section for your operating system:

- “Removing from Solaris”
- “Removing from Windows” on page 88
- “Removing from AIX” on page 88
- “Removing from HP-UX” on page 88
- “Removing from Linux” on page 89

### Removing from Solaris

Complete the following instructions:

1. To remove Tivoli Access Manager for WebSphere, enter the following command:

   ```
   # pkgrm PDWAS
   ```

   A prompt asks you to confirm the removal of the selected package.

2. 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.

3. If you want to remove the supporting Tivoli Access Manager components at this time, use the operating system removal utility to remove the Tivoli Access Manager authorization server (if installed), Access Manager Base Runtime Environment, and Access Manager Java Runtime Environment.
For complete removal instructions, see the IBM Tivoli Access Manager Base Installation Guide.

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

Removing from Windows

Complete the following instructions:
1. Stop and restart the WebSphere Application Server. Click the Add/Remove Programs icon.
2. Select Access Manager for WebSphere.
3. Click Change/Remove.
   The Choose Setup Language dialog box opens.
4. Select a language and click OK.
5. Select the Remove radio button. Click Next.
   The Confirm File Deletion dialog box opens.
6. Click OK.
   The Tivoli Access Manager for WebSphere files are removed.
   The Maintenance Complete dialog box opens.
7. Click Finish.
8. If you want to remove the supporting Tivoli Access Manager components at this time, use the operating system removal utility to remove the Tivoli Access Manager authorization server (if installed), Access Manager Base Runtime Environment, and Access Manager Java Runtime Environment.
   For complete removal instructions, see the IBM Tivoli Access Manager Base Installation Guide.

Removal of Tivoli Access Manager for WebSphere is complete.

Removing from AIX

Use the installp utility to remove the Tivoli Access Manager for WebSphere Application Server AIX package.

If you want to remove the supporting Tivoli Access Manager components at this time, use the operating system removal utility to remove the Tivoli Access Manager authorization server (if installed), Access Manager Base Runtime Environment, and Access Manager Java Runtime Environment. For complete removal instructions, see the IBM Tivoli Access Manager Base Installation Guide.

Removing from HP-UX

Complete the following instructions:
1. To remove Tivoli Access Manager for WebSphere, enter the following command:
   # swremove PDWAS

   A series of status messages appear. A status message indicates that the analysis phase has succeeded. The swremove utility removes the Tivoli Access Manager for WebSphere files from the hard disk.

   When the removal is complete, the swremove utility exits.
2. If you want to remove the supporting Tivoli Access Manager components at this time, use the operating system removal utility to remove the Tivoli Access Manager authorization server (if installed), Access Manager Base Runtime Environment, and Access Manager Java Runtime Environment.

For complete removal instructions, see the IBM Tivoli Access Manager Base Installation Guide.

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

Removing from Linux

Complete the following instructions:

1. To remove Tivoli Access Manager for WebSphere, enter the following command:

   # rpm -e PDWAS-PD

   The files are removed. The rpm utility exits.

2. If you want to remove the supporting Tivoli Access Manager components at this time, use the operating system removal utility to remove the Tivoli Access Manager authorization server (if installed), Access Manager Base Runtime Environment, and Access Manager Java Runtime Environment.

   For complete removal instructions, see the IBM Tivoli Access Manager Base Installation Guide.

Removal of the Tivoli Access Manager for WebSphere package is complete.
Appendix A. Command reference
**pdwascfg**

Configures or unconfigures the Tivoli Access Manager for WebSphere Application Server.

**Syntax**

```
pdwascfg -action {configWAS4 | configWAS5} -remote_acl_user user
-sec_master_pwd password -was_home was_home_dir -pdmgrd_host
policy_serverHostname -pdacld_host authorization_serverHostname
-amwas_home amwas_install_path [-pdmgrd_port policy_server_port] [-pdacld_port
authorization_server_port] [-embedded {true | false}] [-action_type
{all | local | remote}] [-am_domain was_domain] [-cfg_url pdjrte_config_file_URL]
[-key_url pdjrte_keystore_URL] [-verbose {true | false}]
```

```
pdwascfg -action {unconfigWAS4 | unconfigWAS5} -remote_acl_user user
-sec_master_pwd password -was_home was_install_path -pdmgrd_host
policy_serverHostname -pdacld_host authorization_serverHostname
```

```
pdwascfg -help [options]
```

**Parameters**

- **-action {configWAS4 | configWAS5}**
  Specifies the action for this command to perform. Configures the Tivoli Access Manager for WebSphere Application Server.

- **-action {unconfigWAS4 | unconfigWAS5}**
  Specifies the action for this command to perform. Unconfigures the Tivoli Access Manager for WebSphere Application Server.

- **-action_type {all | local | remote}**
  Specifies the level of configuration required. Possible values are: all, local, or remote. The local option performs only configuration changes required on the local machine (meaning no SvrSslCfg). The remote option performs only configuration changes required on the remote machine (meaning SvrSslCfg). The command defaults to all.

- **-am_domain was_domain**
  Specifies the Tivoli Access Manager domain for Tivoli Access Manager for WebSphere. The Tivoli Access Manager authentication server (pdacld) must be in the domain, and the domain must exist in the Tivoli Access Manager protected object space.

- **-amwas_home amwas_install_path**
  Specifies the location of the Tivoli Access Manager for WebSphere installation when Tivoli Access Manager for WebSphere is not installed in the default location. Use this parameter with the -action {configWAS4 | configWAS5} or -action {unconfigWAS4 | unconfigWAS5} options.

  **Note:** The -amwas_home option does not need to be specified as part of the pdwascfg command when Tivoli Access Manager for WebSphere is installed in the default location.

- **-cfg_url pdjrte_config_file_url**
  Specifies the location of the PDJRte properties file. This file is created during configuration and removed during unconfiguration if the option -action_type remote or -action_type all is also specified.
--embedded [true|false]
  Specifies that this product is packaged with WebSphere when set to true. The default value is false.

--help [options]
  Lists the command option name and a short description. If one or more options are specified, it lists each option and a short description.

--key_url pdjrte_keystore_url
  Specifies the location of the PDJrte key store file. This file is created during configuration and removed during unconfiguration if the option -action_type remote or -action_type all is also specified.

--pdacld_host authorization_server_hostname
  Contains the host name of the Tivoli Access Manager authorization server. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

--pdacld_port authorization_server_port
  Specifies the port number of the Tivoli Access Manager authorization server only if it has been configured to be different from the standard port. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options. Note that pdmgrd_port must also be specified if this option is used.

--pdmgrd_host policy_server_hostname
  Contains the host name of the Tivoli Access Manager policy server. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

--pdmgrd_port policy_server_port
  Specifies the port number of the Tivoli Access Manager policy server only if it has been configured to be different from the standard port. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

--remote_acl_user user
  Specifies the user name of the remote acl user. This parameter is used for the SSL connection with the Tivoli Access Manager authorization server. The user should not exist in the registry. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

  For example: --remote_acl_user pdpermadmin

--sec_master_pwd password
  Specifies the password of the administrative user (normally sec_master). Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

--verbose [true|false]
  Enables verbose output when set to true; otherwise, disables verbose output. The default value is false.

--was_home was_home_dir
  Specifies the fully qualified path to the home directory of the WebSphere Application Server installation. Use this parameter with the --action {configWAS4 | configWAS5} or --action {unconfigWAS4 | unconfigWAS5} options.

  For example, c:\WebSphere\AppServer
Comments

The pdwascfg utility is implemented as a shell script on UNIX systems and a batch file on Windows systems. When invoked with action config, the utility completes the following tasks:

- Configures WebSphere to use Tivoli Access Manager for WebSphere.
- Calls the Java class com.tivoli.mts.SvrSslCfg to configure the SSL communication between the Tivoli Access Manager for WebSphere authorization component and both the policy server and the authorization server.
- Creates a user identity for the Tivoli Access Manager for WebSphere classes on the host system.

The script is dependent on finding the correct environment variables for the location of prerequisite software. Set the environment variable %WAS_HOME% to the WebSphere Application Server installation directory. Set %PDWAS_HOME% to the directory location of the Tivoli Access Manager for WebSphere installation directory. The pdwascfg command file calls Java with the following options:

- -Dpdwas.lang.home
  The directory containing the native language support libraries that are provided with Tivoli Access Manager for WebSphere. These are located in a subdirectory under the Tivoli Access Manager for WebSphere installation directory. For example:
  -Dpdwas.lang.home=%PDWAS_HOME%/java
- -Dpdwas.home
  The home (installation) directory for Tivoli Access Manager for WebSphere. For example:
  -Dpdwas.home=%PDWAS_HOME%

Note: This environment variable is set only when a new command window has been opened after installing Tivoli Access Manager for WebSphere.

- -Dwas.home
  The home (installation) directory for WebSphere Application Server. For example:
  -Dwas.home=%WAS_HOME%

Sample Java command, as built by pdwascfg:

java -Dpdwas.lang.home=%PDWAS_HOME%/java
-Dpdwas.home=%PDWAS_HOME%
-Dwas.home=%WAS_HOME%
PDWAScfg -action configWAS
-remote_acl_user ppermadmin
-sec_master_pwd myPassword
-was_home c:/WebSphere/AppServer
-pdmgrd_host pdmgrserver.mysubnet.ibm.com
-pdacld_host pdacldserver.mysubnet.ibm.com

Availability

This command is located in the following default installation directories:

- UNIX systems:
  /opt/amwas/sbin/
- On Windows systems:
  C:\Program Files\Tivoli\amwas\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.
migrateEAR4

Migrates security policy information from deployment descriptors (enterprise archive files) to Tivoli Access Manager for WebSphere Application Server version 4.0.6.

Syntax

migrateEAR4 -j absolute_pathname_to_application_EAR_file -c URI -a admin_ID -p admin_pwd -w WebSphere_admin_ID -d user_registry_domain_suffix [–r root_objectspace_name] [–t ssl_timeout] [–e enterprise_application_name]

Parameters

–a admin_ID
  Specifies the Tivoli Access Manager administrative user. This administrator must have the privileges required to create users, objects, and ACLs. For example, -a sec_master.
  This parameter is optional. When the parameter is not specified, the user is prompted to supply the administrative user name at runtime.

–c URI
  Specifies the Uniform Resource Indicator (URL) location of the PdPerm.properties file that is configured by the pdwascfg utility. When WebSphere Application Server is installed in the default location, the URI is:
  • Solaris, Linux, HP-UX
    file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
  • AIX
    file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
  • Windows
    – WebSphere Application Server 4:
      file:/c:\WebSphere\AppServer\java\jre\PdPerm.properties
    – WebSphere Application Server 5:
      file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

–d user_registry_domain_suffix
  Specifies the domain suffix to be used by the user registry. For example, for LDAP user registries this is the domain suffix, such as:
  "o=ibm,c=us"
  
  Note: Windows requires the domain suffix to be enclosed within quotes.

–e enterprise_application_name
  Specifies the application name so that installed applications, which have a different display name from their installed name, are migrated correctly. If this option is not specified, the utility will attempt to figure out the application name by using either the .ear file or the .xml file.
  An application name can be changed at application deployment or later through the WebSphere console. This change will not be reflected in the EAR file. When the EAR file is not modified to reflect the new name, the wrong protected objects are created. Use the –e option to specify the name of the application as it is displayed on the WebSphere Application Server console.

–j absolute_pathname_to_application_EAR_file
  Specifies the Java 2 Enterprise Edition application archive file. Optionally, this option can also be an EAR directory.
For example, -j /tmp/test_application.EAR

-p admin_prod
Specifies the password for the Tivoli Access Manager administrative user. The administrative user must have the privileges required to create users, objects, and ACLs. For example, you can specify the password for the -a sec_master administrative user as -p myPassword.

This parameter is optional. When it is not specified, the user is prompted to supply the password for the administrative user name.

-r root_objectspace_name
Specifies the root object space name, which is the name of the root of the protected object namespace hierarchy that will be created for WebSphere Application Server. This parameter is optional. The default value for the root object space is WebAppServer.

If a name other than the default is used, the PDWAS.properties file will need to be changed to access the correct object space.

The action group name matches the root object space name. Thus, the action group name is automatically set when the root object space name is specified.

-t ssl_timeout
Specifies the number of minutes for the SSL timeout. This parameter is used to disconnect and reconnect the SSL context between the Tivoli Access Manager authorization server and policy server before the default connection times out.

The default is 60 minutes. The minimum is 10 minutes. The maximum should not exceed the Tivoli Access Manager ssl-v3-timeout value. The default value for ssl-v3-timeout is 120 minutes.

This parameter is optional. If you are not familiar with administration of this value, you can safely use the default value.

-w WebSphere_admin_ID
Specifies the administrative user name that was configured in WebSphere Application Server security user registry field as the administrator. Access as this user is needed to create or update the Tivoli Access Manager protected object space.

When the WebSphere administrative user does not already exist in the protected object space, it is created or imported. In this case, a random password is generated for the user and the account is set to invalid. This password will need to be changed to something known and the account set to valid.

A protected object and ACL are created. The administrative user is added to group pdwas-admin with the following ACL attributes:

- T — traverse permission
- i — invoke permission
- WebAppServer — the action group name. WebAppServer is the default name.

Note that this action group name (and the matching root object space) can be overwritten when the migration utility is run with the -r option.

The group pdwas-admin will need to be added to the admin role if migrating the admin.ear file.
Comments
This utility migrates security policy information from deployment descriptors (enterprise archive files) to Tivoli Access Manager for WebSphere. The utility is implemented as a shell script on UNIX systems and as a batch file on Windows systems. The script calls the Java class com.tivoli.pdas.migrate.Migrate.

The script is dependent on finding the correct environment variables for the location of prerequisite software. The script calls Java with the following options:

- -Dpdwas.lang.home
  The directory containing the native language support libraries that are provided with Tivoli Access Manager for WebSphere. These are located in a subdirectory under the Tivoli Access Manager for WebSphere installation directory. For example:
  -Dpdwas.lang.home=%PDWAS_HOME%\java\nls
- -cp %CLASSPATH% com.tivoli.pdwas.migrate.Migrate
  CLASSPATH must be set correctly for your Java installation.

In addition, on Windows, both the -j option and the -c option can reference the variable %WAS_HOME% to determine where WebSphere Application Server is installed. This information is used to:

- Build the full path name of the enterprise archive file.
- Build the full URI path name to the location of the PdPerm.properties file:

Availability
This command is located in the following default installation directories:

- UNIX systems:
  /opt/amwas/bin/
- On Windows systems:
  C:\Program Files\Tivoli\amwas\bin

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

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.
migrateEAR5

Migrates security policy information from deployment descriptors (enterprise archive files) to Tivoli Access Manager for WebSphere Application Server version 5.0.2.

Syntax

migrateEAR5 -j path -c URI -a admin_ID -p admin_pwd -w Websphere_admin_user -d user_registry_domain_suffix [-r root_objectspace_name] [-t ssl_timeout] [-e enterprise_application_name]

Parameters

-a admin_ID
Specifies the administrative user identifier. The administrative use must have the privileges required to create users, objects, and ACLs. For example, -a sec_master.

This parameter is optional. When the parameter is not specified, the user is prompted to supply the administrative user name at runtime.

-c URI
Specifies the Uniform Resource Indicator (URI) location of the PdPerm.properties file that is configured by the pdwascfg utility. When WebSphere Application Server is installed in the default location, the URI is:

- Solaris, Linux, HP-UX
  file:/opt/WebSphere/AppServer/java/jre/PdPerm.properties
- AIX
  file:/usr/WebSphere/AppServer/java/jre/PdPerm.properties
- Windows
  - WebSphere Application Server 4:
    file:/c:\WebSphere\AppServer\java\jre\PdPerm.properties
  - WebSphere Application Server 5:
    file:"c:\Program Files\WebSphere\AppServer\java\jre\PdPerm.properties"

-d user_registry_domain_suffix
Specifies the domain suffix to be used by the user registry. For example, for LDAP user registries this is the domain suffix, such as:

"o=ibm,c=us"

Note: Windows requires the domain suffix to be enclosed within quotes.

e enterprise_application_name
Specifies the application name so that installed applications, which have a different display name from their installed name, are migrated correctly. If this option is not specified, the utility will attempt to figure out the application name by using either the .ear file or the .xml file.

An application name can be changed at application deployment or later through the WebSphere console. This change will not be reflected in the EAR file. When the EAR file is not modified to reflect the new name, the wrong protected objects are created. Use the -e option to specify the name of the application as it is displayed on the WebSphere Application Server console.
-j path
   Specifies the fully qualified path and file name of the Java 2 Enterprise Edition
   application archive file. Optionally, this path can also be a directory of an
   expanded enterprise application.

   For example, -j /tmp/test_application.EAR

-p admin_pwd
   Specifies the password for the Tivoli Access Manager administrative user. The
   administrative user must have the privileges required to create users, objects,
   and ACLs. For example, you can specify the password for the -a sec_master
   administrative user as -p myPassword.

   This parameter is optional. When it is not specified, the user is prompted to
   supply the password for the administrative user name.

-r root_objectspace_name
   Specifies the root object space name that is the name of the root of the
   protected object namespace hierarchy that will be created for WebSphere
   Application Server. This parameter is optional.

   The default value for the root object space is WebAppServer. If a name other
   than the default is used, then the PDWAS.properties file will need to be
   changed to access the correct object space.

   The action group name matches the root object space name. Thus, the action
   group name is automatically set when the root object space name is specified.

-t ssl_timeout
   Specifies the number of minutes for the SSL timeout. This parameter is used to
   disconnect and reconnect the SSL context between the Tivoli Access Manager
   authorization server and policy server before the default connection times out.

   The default is 60 minutes. The minimum is 10 minutes. The maximum should
   not exceed the Tivoli Access Manager ssl-v3-timeout value. The default value
   for ssl-v3-timeout is 120 minutes.

   This parameter is optional. If you are not familiar with administration of this
   value, you can safely use the default value.

-w WebSphere_admin_user
   Specifies the user name that was configured in the WebSphere Application
   Server security user registry field as the administrator. Access permission for
   this user is needed to create or update the Tivoli Access Manager protected
   object space.

   When the WebSphere administrative user does not already exist in the
   protected object space, it is created or imported. In this case, a random
   password is generated for the user and the account is set to invalid. This
   password will need to be changed to something known and the account set to
   valid.

   A protected object and ACL are created. The administrative user is added to
group pdwas-admin with the following ACL attributes:
   • T — traverse permission
   • i — invoke permission
   • WebAppServer— the action group name. WebAppServer is the default name.

   Note that this action group name (and the matching root object space) can
   be overwritten when the migration utility is run with the -r option.
Add the group pdwas-admin to the administrator role if you are migrating the adminconsole.ear file.

**Comments**

This utility migrates security policy information from deployment descriptors (enterprise archive files) to Tivoli Access Manager for WebSphere. The utility is implemented as a shell script on UNIX systems and as a batch file on Windows systems. The script calls the Java class com.tivoli.pdas.migrate.Migrate.

The script is dependent on finding the correct environment variables for the location of prerequisite software. The script calls Java with the following options:

- `-Dpdwas.lang.home`
  
  The directory containing the native language support libraries that are provided with Tivoli Access Manager for WebSphere. These are located in a subdirectory under the Tivoli Access Manager for WebSphere installation directory. For example:
  
  - `Dpdwas.lang.home=%PDWAS_HOME%\java\nls`

- `-cp %CLASSPATH% com.tivoli.pdwas.migrate.Migrate`
  
  CLASSPATH must be set correctly for your Java installation.

In addition, on Windows, both the `-j` option and the `-c` option can reference the variable `%WAS_HOME%` to determine where WebSphere Application Server is installed. This information is used to:

- Build the full path name of the enterprise archive file.
- Build the full URI path name to the location of the PdPerm.properties file.

**Availability**

This command is located in the following default installation directories:

- UNIX systems:
  
  `/opt/amwas/bin/`

- On Windows systems:
  
  `C:\Program Files\Tivoli\amwas\bin`

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

**Return Codes**

The following exit status codes can be returned:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The command completed successfully.</td>
</tr>
<tr>
<td>1</td>
<td>The command failed.</td>
</tr>
</tbody>
</table>

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 B. 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.
Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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 manager 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.
cipher. Encrypted data that is unreadable until it has been converted into plain data (decrypted) with a key.

customize programming. WebSEAL.

configuration.  (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (2) The machines, devices, and programs that make up a system, subsystem, or network.

connection. (1) In data communication, an association established between functional units for conveying information. (2) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In the Internet, a connection extends from a TCP application on one system to a TCP application on another system. (3) In system communications, a line over which data can be passed between two systems or between a system and a device.

container object. A structural designation that organizes the object space into distinct functional regions.

cookie. Information that a server stores on a client machine and accesses during subsequent sessions. Cookies allow servers to remember specific information about clients.

credentials. Detailed information, acquired during authentication, that describes the user, any group associations, and other security-related identity attributes. Credentials can be used to perform a multitude of services, such as authorization, auditing, and delegation.

credentials modification service. An authorization API runtime plug-in which can be used to modify a Tivoli Access Manager credential. Credentials modification services developed externally by customers are limited to performing operation to add and remove from the credentials attribute list and only to those attributes that are considered modifiable.

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.

digital signature. In e-commerce, data that is appended to, or is a cryptographic transformation of, a data unit and that enables the recipient of the data unit to verify the source and integrity of the unit and to recognize potential forgery.

DN. See distinguished name.

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.

E

EAS. See External Authorization Service.

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.

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.

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**


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 work 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.

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.
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.

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.

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 control lists 5
access setting
  inheritance 7
  overriding 7
admin.ear 32, 34, 43, 75, 84
administration API 8, 52
administration tools
  pdadmin 52
Web portal manager 52
administrative user
  WebSphere
    creating 26
Advanced Edition 3
amwas_migrate.log 46, 77, 85
authorization decision 3
authorization server
  configuring additional 55

B
back-up 67
bulk loading of users 15

C
cache tables
  specifying 53
CLASSPATH
  setting 31
com.tivoli.mts.SvrSslCfg 31
configuration parameters
  specifying 52
 configuring
  additional systems 25, 39
  dynamic role caching 53
  initial system 25
  Tivoli Access Manager 25
configWAS4 30
configWAS5 30
creating
  WebSphere administrative user 26

D
declarative security 2
deployment descriptors 3, 5, 43, 47
Directory 15
disk requirements 12
domain, joining 30
DTD 43
dynamic role caching
  configuring 53

E
EAR files 47

G
groups 4
GSO principal mapping 57

H
host-based security 15, 51

I
importing users 15
installation packages 11
Installing Tivoli Access Manager
  AIX 20
  HP-UX 21
  Linux 22
  Solaris 19
  Windows 23
installp 20
InstallShield 23
introduction 1

J
J2EE security 4
Java property file 52
Java Runtime Environment
  prerequisite versions 14

L
logging 61

M
mapping
  groups to roles 4
  principals to roles 4
  user to roles 45
maximum number of users
  specifying 53
memory requirements 12
migrateEAR4 utility 96
migrateEAR5 utility 99
migrating security settings
  WebSphere version 4.0.6 31
  WebSphere version 5.0.2 34
migration utility
  initial use 33, 46, 77, 85
  limitations 47
logging 33, 46, 77, 85
overview 43
tutorial 75, 84
using 43

© Copyright IBM Corp. 2002, 2003
operating systems
  supported versions  11

P
padmin
  adding users  70, 79
  changing roles  77, 85
  creating administrative user  26
  importing users  15
  summary  8
PDPerm.properties  32, 34, 43, 75, 84
PDWAS_HOME, setting  30
pdwas_migrate.log  33
pdwas-admin group
  adding to ACL  33
PDWAS.properties  52
pdwascfg
  using  30, 41
  pdwascfg utility  92
pkgadd  19
policy management
  centralizing  8
  principal life time
  specifying  53
  principals  4
problem determination  47, 65
programmatic security  2

R
related publications  x
removing
  AIX  88
  HP-UX  88
  Linux  89
  Solaris  87
  Windows  88
removing Tivoli Access Manager  87
requirements
  disk space  12
  memory  12
Role Based Policy Framework Parameters
  specifying  54
  role lifetime
  specifying  53
  role mappings  45
  role-based security  4
roles  4
rpm  22, 89

S
security
  declarative  2
  programmatic  2
setupCmdLine  30
Single Server
  adding users to ACLs  46
  host-based security  51
  single sign-on with WebSEAL  63
software contents  11
software prerequisites
  Tivoli Access Manager Base  13
ssl timeout value  47
static role caching
  enabling  52
static roles
  defining  52
supported platforms  11
SvrSslCfg
  adding authorization server  55
swinstall  21
swremove  88

T
Tivoli Access Manager
  administration API  8, 52
  authorization framework  1
  authorization server  9
  configuring  25
  integrating with WebSphere  2
  policy database
    replicating  9
  policy server  5, 13
  removing  87
  secure domain  13
  security model  1
  trace  61
troubleshooting tips  47, 65
tutorial
  adding security  71, 80
  adding user to LDAP  70, 79
  changing roles  77, 85
  enabling security  69, 78
  enabling security roles  67
  overview  69, 78
  testing security
    Servlet  75, 83
    Thick Client  75, 83
  using migration utility  75, 84

U
upgrading
  from Access Manager 3.9  15
  from Policy Director 3.8  15
user registry
  LDAP  15
  prerequisites  14
  shared  3, 12
users
  mapping to roles  45
utilities
  migrateEAR4  96
  migrateEAR5  99
  pdwascfg  92

W
WAS_HOME
setting  31, 34
Web portal manager  8, 52
WebSphere
  Admin Client  74, 82
  administration server  27
  Advanced Edition
    Single Server  12, 51
    Bindings  72
WebSphere (continued)
documentation URL 12
EJB Modules 72
managing multiple servers 8
Security Constraints 72, 81
Security Roles 72
Single Server
host-based security 15
migration steps 46
tutorial
deploying application 74, 82
Web Modules 72, 81
WebSphere Advanced Edition 3
WebSphere console
configuring Tivoli Access Manager groups 56
WebSphere security
enabling in version 4.0.6 27
enabling in version 5.0.2 27
WebSphere v5.1
configuring 36