Sun Java System Access Manager 7.1 Deployment Planning Guide

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

Part No: 819–4672–10
March 2007

Products covered by and information contained in this publication are controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical or biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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

Les produits qui font l'objet de cette publication et les informations qu'il contient sont régis par la legislation américaine en matière de contrôle des exportations et peuvent être soumis au droit d'autres pays dans le domaine des exportations et importations. Les utilisations finales, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes chimiques ou biologiques ou pour le nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers des pays sous embargo des Etats-Unis, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exclusive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la legislation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement designés, sont rigoureusement interdites.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFACON.

070223@16599

Contents

Tables

Figures

Preface

The *Sun Java System Access Manager 7.1 Deployment Planning Guide* provides planning and deployment solutions for Sun Java™ System Access Manager based on the solution life cycle.

Access Manager is a component of the Sun Java Enterprise System (Java ES), a set of software components that provide services needed to support enterprise applications distributed across a network or Internet environment.

Who Should Use This Book

This book is intended for deployment architects and business planners responsible for the planning, analysis, and design of an Access Manager deployment. This book might also be useful for system integrators who are responsible for the design and implementation of the specific aspects of an Access Manager deployment.

Before You Read This Book

Readers should be familiar with the following components and concepts:

- Access Manager technical concepts, as described in the *Sun Java System Access Manager 7.1 Technical Overview*
- Deployment platform: Solaris™, Linux, HP-UX, or Windows operating system
- Web container that will run Access Manager: Sun Java System Application Server, Sun Java System Web Server, BEA WebLogic, or IBM WebSphere Application Server
- Technical concepts: Lightweight Directory Access Protocol (LDAP), Java technology, JavaServer Pages™ (JSP) technology, HyperText Transfer Protocol (HTTP), HyperText Markup Language (HTML), and eXtensible Markup Language (XML)

How This Book Is Organized

This guide is based on the solution life cycle, which describes the various phases of deployment planning. Chapter 1 provides a description of the solution life cycle.

Related Books

Related documentation is available as follows:

- "Access Manager 7.1 Documentation Set" on page 12
- "Sun Java Enterprise System 5 Documentation" on page 13

Access Manager 7.1 Documentation Set

The following table describes the Access Manager documentation set, which is available on the following web site:

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

TABLE P–1 Access Manager 7.1 Documentation Set

Title	Description
Sun Java System Access Manager 7.1 Documentation Center	Provides links to commonly referenced information in the Access Manager 7.1 documentation collection.
Sun Java System Access Manager 7.1 Release Notes	Describes new features, problems fixed, installation notes, and known issues and limitations. The Release Notes are updated periodically after the initial release to describe any patches, new features, or problems.
Sun Java System Access Manager 7.1 Technical Overview	Explains basic Access Manager concepts and terminology and provides an overview of how Access Manager components work together to consolidate access control functions and to protect enterprise assets and web-based applications.
Sun Java System Access Manager 7.1 Deployment Planning Guide (this guide)	Provides planning and deployment solutions for Access Manager based on the solution life cycle.
Sun Java System Access Manager 7.1 Postinstallation Guide	Provides information about configuring Access Manager after installation. Usually, you perform postinstallation tasks only a few times. For example, you might want to deploy an additional instance of Access Manager or configure Access Manager for session failover.

TABLE P–1 Access Manager 7.1 Documentation Set *(Continued)*

Title	Description
Sun Java System Access Manager 7.1 Administration Guide	Describes various administrative tasks such as realms management, policy management, authentication, and directory management.
Sun Java System Access Manager 7.1 Administration Reference	Provides reference information for the Access Manager command-line interface (CLI), configuration attributes, `AMConfig.properties` attributes, `serverconfig.xml` file attributes, log files, and error codes.
Sun Java System Access Manager 7.1 Federation and SAML Administration Guide	Provides information about Federation Manager based on the Liberty Alliance Project specifications. It includes information on the integrated services based on these specifications, instructions for enabling a Liberty-based environment, and summaries of the application programming interface (API) for extending the framework.
Sun Java System Access Manager 7.1 Developer's Guide	Provides information about customizing Access Manager and integrating its functionality into an organization's current technical infrastructure. It also contains details about the programmatic aspects of the product and its API.
Sun Java System Access Manager 7.1 C API Reference	Provides summaries of data types, structures, and functions that make up the Access Manager public C APIs.
Sun Java System Access Manager 7.1 Java API Reference	Provides information about the implementation of Java packages in Access Manager.
Sun Java System Access Manager 7.1 Performance Tuning Guide	Provides information about how to tune Access Manager and its related components for optimal performance.
Sun Java System Access Manager Policy Agent 2.2 User's Guide	Provides an overview of Policy Agent software, including the web agents and J2EE agents that are currently available. To view the Access Manager Policy Agent 2.2 documentation collection, see: `http://docs.sun.com/coll/1322.1`

Sun Java Enterprise System 5 Documentation

The following table provides links to documentation collections for related Java ES products.

TABLE P–2 Related Sun Java Enterprise System 5 Documentation

Product	Link
Sun Java Enterprise System 5	`http://docs.sun.com/coll/1286.2`

TABLE P–2 Related Sun Java Enterprise System 5 Documentation *(Continued)*

Product	Link
Sun Java System Directory Server Enterprise Edition 6.0	`http://docs.sun.com/coll/1224.1`
Sun Java System Web Server 7.0	`http://docs.sun.com/coll/1308.3`
Sun Java System Application Server Enterprise Edition 8.2	`http://docs.sun.com/coll/1310.3`
Sun Java System Message Queue 3.7 UR1	`http://docs.sun.com/coll/1307.2`
Sun Java System Web Proxy Server 4.0.4	`http://docs.sun.com/coll/1311.4`
Sun Java System Identity Manager 7	`http://docs.sun.com/coll/1514.2`

Searching Sun Product Documentation

Besides searching Sun product documentation from the docs.sun.com^{SM} web site, you can use a search engine by typing the following syntax in the search field:

search-term `site:docs.sun.com`

For example, to search for “broker,” type the following:

```
broker site:docs.sun.com
```

To include other Sun web sites in your search (for example, java.sun.com, www.sun.com, and developers.sun.com), use `sun.com` in place of `docs.sun.com` in the search field.

Related Third-Party Web Site References

Third-party URLs are referenced in this document and provide additional, related information.

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

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Typographic Conventions

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

TABLE P–3 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P–4 Shell Prompts

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

Sun Welcomes Your Comments

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

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

For example, the title of this book is *Sun Java System Access Manager 7.1 Deployment Planning Guide*, and the part number is 819-4672.

CHAPTER 1

Introduction to Deployment Planning for Access Manager

Sun Java™ System Access Manager (Access Manager) is part of the Sun Identity Management infrastructure that allows an organization to manage secure access to web applications and other resources both within an enterprise and across business-to-business (B2B) value chains. This chapter introduces the basic Access Manager deployment planning principles, including:

- "About Access Manager" on page 17
- "Access Manager Deployment Planning" on page 19

About Access Manager

Access Manager is a component of Sun Java Enterprise System (Java ES), a set of software components that provide services that support enterprise applications distributed across a network or Internet environment. Access Manager provides these major functions:

- Centralized authentication and authorization services using both role-based and rule-based access control
- Single sign-on (SSO) for access to an organization's web-based applications
- Federated identity support with the Liberty Alliance Project and Security Assertions Markup Language (SAML)

- Logging of critical information including administrator and user activities by Access Manager components for subsequent analysis, reporting, and auditing. Logging is based on the J2SE logging APIs (`java.util.logging`).

Access Manager is also part of the Sun Identity Management Suite, which provides the functions required to use, share, and manage identity information, including directory services, access management, provisioning, and federation. The products in the Identity Management Suite include:

- Sun Java System Access Manager
- Sun Java System Directory Server Enterprise Edition
- Sun Java System Federation Manager
- Sun Java System Identity Manager

For more information about each component, see the Sun Software web site: http://www.sun.com/software/

The following figure shows the Access Manager, Identity Manager, and Directory Server identity management components.

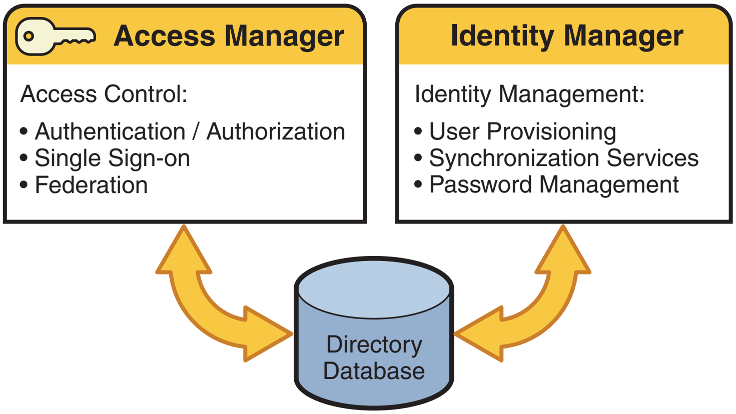


FIGURE 1–1 Sun Identity Management Components

Sun Java System Identity Manager provides user provisioning, password management, synchronization services, comprehensive audit and reporting, and delegated administration. Identity Manager is not a component of Sun Java Enterprise System. To use Identity Manager in your deployment or to obtain more information, contact your Sun Microsystems technical representative or a Sun sales office: http://www.sun.com/sales-n-service/WWSales.jsp.

For a detailed description of Access Manager, see the *Sun Java System Access Manager 7.1 Technical Overview*.

Access Manager Deployment Planning

Deployment planning is a critical step in the successful implementation of an identity management solution. Each enterprise has its own set of goals, requirements, and priorities to consider. Successful deployment planning is the result of careful preparation, analysis, and design. Errors and missteps that occur anywhere during the planning process can result in a system that can misfire in many ways. Significant problems can arise from a poorly planned system. For example, the system could under-perform, be difficult to maintain, be too expensive to operate, could waste resources, or could be unable to scale to meet increasing needs.

Access Manager deployment planning as described in this guide follows the solution life cycle. The solution life cycle includes the process of planning, designing, and implementing an Access Manager enterprise software solution based on Java Enterprise System.

Solution Life Cycle

The solution life cycle, shown in the following figure, is a useful tool for planning and tracking a deployment project. The life cycle structures the preparation, analysis, and design necessary for successful deployment planning into a series of ordered phases. Each phase consists of related tasks that result in outputs that are carried forward as inputs to subsequent phases. The tasks within each phase are iterative, requiring thorough analysis and design before generating the outputs for that phase.

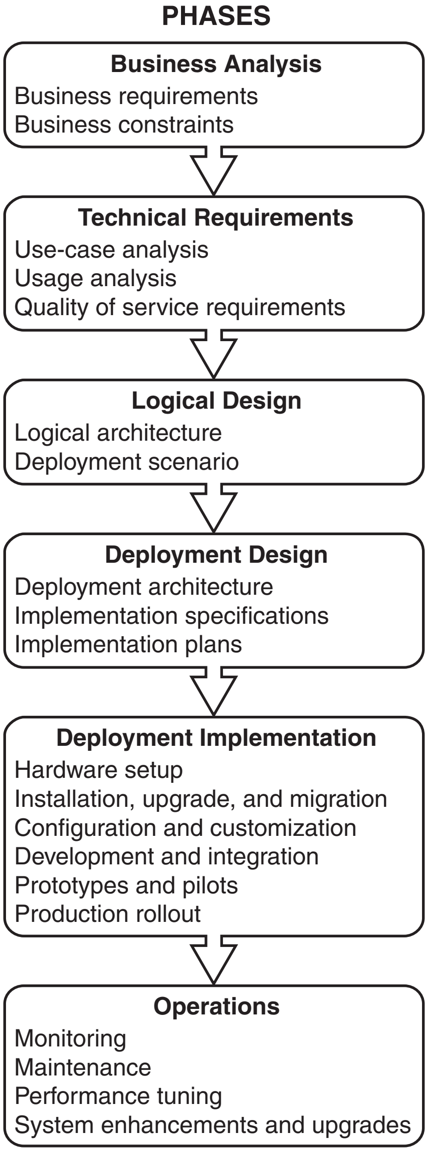


FIGURE 1–2 Solution Life Cycle

The organization of this manual is based on phases within the solution life cycle. The following sections in this chapter briefly describe each life cycle phase. For a more detailed description of these phases, see the *Sun Java Enterprise System 2006Q3 Deployment Planning Guide*.

Business Analysis Phase

During business analysis, you define the business goals of a deployment project and state the business requirements that must be met to achieve those goals. When stating the business requirements, consider any business constraints that might affect the ability to achieve the business goal. Without proper business analysis, you run the risk of an incomplete solution.

During the business analysis phase you create business requirements documents that you later use as inputs to the technical requirements phase.

See Chapter 2.

Technical Requirements Phase

The technical requirements phase starts with the business requirements and business constraints defined during the business analysis phase and translates them into technical specifications that can be used to subsequently design the deployment architecture. The technical requirements specify quality of service (QoS) features, such as performance, availability, security, and others.

During the technical requirements phase, you create documents that contain the following information:

- Analysis of user tasks and usage patterns
- Use cases that model user interaction with the planned system
- Quality of service requirements derived from the business requirements, possibly taking into consideration the analysis of user tasks and usage patterns

The resulting usage analysis, use cases, and QoS requirements documents are inputs to the logical design phase of the solution life cycle. The usage analysis also plays a significant role in the deployment design phase.

See Chapter 3.

Logical Design Phase

During logical design, using use cases from the technical requirements phase as inputs, you identify the Access Manager components necessary to implement a solution. You also identify components that provide support to those Java ES components, and any additional custom-developed components necessary to meet the business requirements. You then map the components within a logical architecture that shows the interrelationships among the components. The logical architecture does not specify any hardware required to implement the solution.

The output of the logical design phase is the logical architecture. The logical architecture and the QoS requirements from the technical requirements phase form a deployment scenario, which is the input to the deployment design phase.

See Chapter 4.

Deployment Design Phase

During deployment design, you map the components specified in the logical architecture to a physical environment, producing a high-level deployment architecture. You also create an implementation specification, which provides low-level details specifying how to build the deployment architecture. Additionally, you create a series of plans and specifications that detail different aspects of implementing the software solution.

Project approval occurs during the deployment design phase. During project approval, the cost of the deployment is assessed. If approved, contracts for implementation of the deployment are signed, and resources to build the project are acquired. Often, project approval occurs after the implementation specification has been detailed. However, approval can also occur upon completion of the deployment architecture.

The outputs of the deployment design phase include the following:

- **Deployment architecture.** A high-level design document that represents the mapping of components to network hardware and software.
- **Implementation specifications.** Detailed specifications used as blueprints for building the deployment.
- **Implementation plans.** A group of plans and specifications that cover various aspects of implementing an enterprise software solution. Implementation plans include a migration plan, installation plan, user management plan, test plan, and others.

See Chapter 5

Implementation Phase

During the implementation phase, you work from specifications and plans created during deployment design to build the deployment architecture and implement the solution.

See Chapter 6.

◆ ◆ ◆ CHAPTER 2

Business Analysis for Access Manager

Sun Java™ System Access Manager allows an organization to deploy an identity management solution for employees, contractors, customers, and suppliers. During the business analysis phase of the solution life cycle, you define business goals by analyzing a business problem and identifying the business requirements and business constraints to meet that goal. This chapter contains the following sections:

- "About Business Analysis" on page 23
- "Defining Access Manager Business Requirements" on page 24
- "Setting Goals" on page 28
- "Gathering Information" on page 29
- "Evaluating Applications" on page 31
- "Categorizing Data" on page 33
- "Building a Time Line" on page 36

About Business Analysis

Business analysis starts with stating the business goals. You then analyze the business problems you must solve and identify the business requirements that must be met to achieve the business goals. Consider also any business constraints that limit your ability to achieve the goals. The analysis of business requirements and constraints results in a set of business requirements documents.

You use the resulting set of business requirements documents as a basis for deriving technical requirements in the technical requirements phase. Throughout the solution life cycle, you measure the success of your deployment planning and ultimately the success of your solution according to the analysis performed in the business analysis phase.

Defining Access Manager Business Requirements

This section provides specific business requirements to consider for Access Manager (that is, which business requirements imply a need for an Access Manager solution).

Sun Java System Access Manager is a complex, distributed identity management system that, when properly deployed, secures access to a wide variety of data and services spanning an enterprise's organizations. To ensure proper control over corporate resources, appropriate planning of the deployment process is required. This chapter offers information about how to plan the deployment, including:

- "Defining Resources" on page 24
- "Independent Software Vendors" on page 27
- "Third Party Affiliates" on page 27
- "Funding" on page 28

Defining Resources

Because an identity management solution involves a broad variety of systems throughout an organization, proper Access Manager deployment requires a variety of resources. The following corporate resources will be involved or required in the deployment process.

- "Human Resources" on page 24
- "Executive Sponsors" on page 25
- "Team Lead" on page 25
- "Project Management" on page 25
- "Systems Analyst" on page 26
- "Line-of-Business (LOB) Application Administrators" on page 26
- "System Administrators" on page 26

Human Resources

You should consider the various business and political relationships within an organization. A team of individuals should be assembled with a direct or matrixed reporting structure. Typically, Access Manager deployments have small teams that might consist of a project manager and several dedicated System Administrators. These people report to the Team Lead and further up to an owner who has responsibility across a number of related projects and often reports directly to an executive sponsor. This group is often augmented by virtual team members consisting of Sun technical resources, and LOB Application Administrators, which are used as required.

While this structure might not meet your exact needs, it does represent a fairly typical deployment team model. Although not necessarily distinct individuals, the following abstract technical roles representing various skill sets further define a typical Access Manager deployment team.

Executive Sponsors

Successful identity management deployments traditionally cross organizational and political boundaries, which requires buy-in and support from those setting direction for the company. It is critical that executive sponsorship be in place. Planning meetings are an important process for gaining insight from those with a vested interest in the deployment. As the project plan is developed, ensure that its deliverables are inline with the goals of the company as a whole. For example, if cost reduction is a core business driver, collect statistics on current identity management costs and then determine costs such as using the help desk for password resets? Having tangible statistics available can help define a specific return on investment (ROI) as the deployment team attempts to gain executive support. Other company issues that might be relevant include:

- Who benefits from the identity management deployment?
- What organizational problems does an identity management solution solve?
- How does the company address internal issues that might slow the deployment?

Often the identity management concepts and the value of an Access Manager deployment must be related to other executives. A business and technology evangelist can sell the new infrastructure to executives, helping to drive the demand for integration and aid in the acceptance and ultimate success of the infrastructure changes.

Team Lead

A team lead should be chosen as the party responsible for the project's success. The team lead must be in charge and have the authority to make the project's goals happen. The team lead might be a logically distributed role, perhaps between a technical lead, a project manager, and an executive. However you define this role, the goal is to show continued progress and demonstrated success throughout the deployment process to maintain executive sponsorship.

Project Management

A project manager is responsible for the coordination of schedules. The project manager maintains a schedule that correlates the availability of services, support provided by the core IT group and the integration of the various line-of-business (LOB) applications. This person must have strong communication skills and understand the political aspects of the company. The project manager must also balance the needs of the internal customers with the availability of resources in order to support new applications joining the environment.

LOB applications are vital to running an organization. They are generally large programs with capabilities that tie into databases and database management systems. They can include accounting, supply chain management, and resource planning applications. Increasingly, LOB applications are being connected with network applications that have user interfaces and with personal applications such as e-mail and address books.

Systems Analyst

A systems analyst is responsible for assessment and categorization of the various data and services to be integrated into the Access Manager deployment. The systems analyst interviews the LOB application owners and gathers details on technical requirements including platform, architecture, and the deployment schedule. With this information, the systems analyst formulates a plan about how the application will be integrated into the deployment in order to meet their customer's requirements. The systems analyst must be an IT generalist, with broad knowledge of various application architectures and platforms. Detailed knowledge of Access Manager architecture, services, agents, and APIs is also required.

Line-of-Business (LOB) Application Administrators

LOB application administrators are technical specialist with intimate knowledge of, and control over, the LOB application and are responsible for integration of the Access Manager policy agents, or policy enforcement point, into their application. They must clearly communicate the LOB application's architecture, its integration points, and appropriate schedules. They are typically responsible for defining the access control model represented in Access Manager policies. They might perform custom programming to enhance the integration between Access Manager and their application (for example, session coordination). Finally, they are generally responsible for quality assurance (QA) and the regression testing of their application within the newly-deployed environment.

System Administrators

It is critical that appropriate resources are in place to deploy and maintain the availability of Access Manager. System administrators are required at the following levels. Additional administrators might also include a web container administrator who is responsible for the deployment and performance of the software container in which Access Manager is deployed.

Access Manager Administrator

The Access Manager administrator is responsible for the deployment and maintenance of Access Manager. This administrator assures the availability of the common services, provides necessary enhancements to the infrastructure in general, and configures policies and roles in particular. This administrator also helps support integration efforts by developing guidelines, and offers technical support to the LOB application administrators. An understanding of Java, XML, LDAP, HTTP, and web application architectures is critical.

Directory Server Administrator

Corporate directory services used for authentication and authorization are often already managed by a group within the organization before the Access Manager deployment is even considered. The Directory Server administrator is responsible for the availability of the directory services, as well as for accepting and integrating additions or modifications to the currently defined LDAP schema and identity data, including changes that are required to support the identity management infrastructure.

Hardware, Datacenter, and Network Administrator

Large organizations typically find economies of scale by separating hardware, operating system, data center, and network administration from middleware administration. If this is the case in your company, it is essential that there is clear communication between these various administrators. It may be critical to the deployment's success to have access to certain machines or to establish certain network configurations; keeping these administrators aware of project milestones and requirements can facilitate a smooth rollout.

Independent Software Vendors

Sun Microsystems and other independent software vendors (ISV) are critical partners in the successful deployment of Access Manager. Purchasing packaged software allows an enterprise to diminish and distribute the cost and risk of software development across multiple organizations.

An ISV makes and sells software products that can run on one or more types of computer hardware or operating system platforms. The companies that make the platforms (for example, Sun, IBM, Hewlett-Packard, Apple, or Microsoft) encourage and lend support to the ISV.

It is in the best interest of all parties involved for ISV to develop cooperative relationships and drive successful deployments. Engage Sun technical services and other ISVs to help bootstrap the project and to convey knowledge they have gained from previous Access Manager deployments. Using technical services, as well as an open discussion with your account team (who can act as an intermediary between Access Manager engineers and your deployment team) can help insure your investment and a successful deployment.

Third Party Affiliates

If you are planning on leveraging the Federation Management capabilities of Access Manager, you will be collaborating with external partners and third party affiliates. Consider an initial deployment of this functionality in conjunction with your own internal deployment. In this case, it is important to involve the LOB application that owns the business functionality that will be delivered and to maintain communication with the technical resources of all parties. Your legal counsel can also help to establish a good relationship between involved parties.

Funding

The core IT group is often responsible for the cost of the deployment project. In fact, it is common to have internal funds transferred from an LOB application to the core group in order to fund portions of the identity management project. But, even when a single LOB application group is providing initial funding, the needs of the larger organization should be balanced with the needs of the funding group.

Setting Goals

By setting goals, an organization defines where it wants to be after the Access Manager deployment is finished. The deployment strategy is to plan a roadmap for reaching these objectives and move towards it. Goals are created by defining the expectations of all involved and getting approval early in the process.

In general, an identity management solution enhances security and improves infrastructure manageability while decreasing costs. More specifically, some common goals (and their benefits) that Access Manager allows an organization to meet include:

- Implementation of a scalable infrastructure to meet the expected increase in digital identities (employees, partners, and customers).
- Consolidation of the creation and management of identity profiles with each group controlling their own data.
- Cost reduction through vendor consolidation, user self-management and related administration costs.
- Improvement of security through immediate identity profile termination.
- Improved transparency of security models and access rights.
- Condensing time required for access to critical systems.
- Removal of user rights to critical systems as roles or affiliation within the organization change.

Ultimately, these goals, combined with an understanding of the motivation of all groups involved and information gleaned from a site survey, can be used to design an infrastructure for the deployment. In addition, they can be used throughout the deployment process to keep interested parties engaged and encourage project endorsement.

Gathering Information

A site survey can be used to gather information about the applications and data stores that will be integrated into the deployment. In addition, these departmental interviews help to forge an understanding of the motivation of the groups involved by defining their particular functions and goals. Once collected, the information can solidify buy-in from the executive sponsors as well as serve as a design blueprint. The following groups of individuals can help in a site survey:

- Users provide feedback about the applications they use on a daily basis.
- Human resources provides information about hiring and termination processes.
- Support personnel offer insight into problems that cross organizational boundaries.
- Application administrators and developers can provide technical information about the line-of-business (LOB) applications to be integrated into the deployment.
- Network administrators have knowledge of the organization's technical baseline for performance and standards.

An initial survey might include gathering information about the following items:

- "Business Processes" on page 29
- "IT Infrastructure" on page 30
- "Virtual Data" on page 30

Business Processes

The business processes are the procedures that diverse groups in the organization define to do their job. Processes can include procedures for:

- Issuing payroll
- Purchasing and accounts payable
- Authorizing employee travel
- Departmental budgeting
- Terminating employees

It is imperative to assess these processes because they are generally supported by the applications used by each business unit. Things to consider include:

- Do the current processes cause delays?
- Are there a number of different processes that perform the same function?
- Can processes be standardized across business unit boundaries?
- How complex are the processes? Can they be consolidated or simplified?
- Can the current processes handle organizational changes?

Any changes to be made to the processes should be initiated prior to the beginning of the deployment.

IT Infrastructure

The IT infrastructure includes all the hardware servers, operating systems, and integrated applications that will be integrated into the Access Manager deployment. Consider the following:

- What applications will leverage Access Manager?

  Applications might include critical internal applications such as those for human resources and accounting or less-critical employee portals. Also leveraging the functionality of Access Manager might be external business-to-business applications that deal with both confidential financial information and less confidential sales material, or business-to-consumer shopping carts that are concerned with credit card data and purchase histories.
- What systems will leverage Access Manager?

  Consider the hardware on which applications are being deployed as well as their operating systems. An Access Manager deployment, at the minimum, includes a web container to run the application, a Sun Java System Directory Server (or existing data store), and Access Manager. Additional hardware servers might run their own web containers with corporate resources and on which Access Manager policy agents can be installed for improved security purposes.
- What Access Manager services will each department leverage?

  Consider the default and custom services integrated within Access Manager. Role and policy strategies will have to be mapped and defined for each department. Authentication modules need to be assessed and custom services, if any, need to be developed.

Other technical considerations also include:

- Are there incompatibilities in the infrastructure?
- Does the current system experience slowdowns or down time?
- Are the applications sufficiently secure?
- Are there virus control procedures?
- Can applications be customized based on user entitlements?

For more information, see “Evaluating Applications” on page 31.

Virtual Data

Virtual data is a catch-all phrase for the profiles that will access, the configurations that will be accessible from, and the data that will be secured by Access Manager. Virtual data includes, but is not limited to, user profiles (such as employees or customers), data and service access rules, and other types of corporate data.

- What assets will Access Manager be protecting?

Access Manager secures access to all types of data and services. An administrator can regulate who can view or configure Access Manager data as well as control access to applications, portals, and services.

- What users will leverage Access Manager?

  Users might include employees, business partners, suppliers, and current or potential customers. Each user will have a profile that includes, at a minimum, their user ID and password. Employees will undoubtedly have larger and more confidential profiles than customers who access external sales information.

- What data will be accessible?

  Data might include public information, internal information, confidential information, and restricted data. Data might also include sales information on an external web site, confidential employee profiles, access rules that protect corporate resources, server configuration information, and federated customer profiles.

- What is the authoritative source of the data?

  Often multiple schemas that define different types of data are used. These definitions need to be reconciled within your deployment. Be aware of data ownership issues, allowing the various LOB applications to maintain control over their data, where appropriate. It is imperative to balance the demands of the satellite groups in order to provide service that is representative of the overall enterprise as all services are critical to the larger organization.

Other technical considerations also include:

- Is the same information defined in multiple attributes?
- Do users have multiple cross-organizational profiles?
- Are the data stores located in front of the firewall?
- Is the data consistent across different data stores?
- How often is new data added or existing data modified?

For more information, see “Categorizing Data” on page 33.

Evaluating Applications

Identity management services are generally provided as a centralized IT function with corporate and business unit applications forming the extended system. Upkeep of this system hierarchy involves a core IT group that manages and maintains the server infrastructure and a satellite group of employees to maintain the LOB applications.

As large organizations often have hundreds (or even thousands) of deployed internal applications, evaluating all of them would be time-intensive and cost-prohibitive. When conducting an application survey, focus on applications that meet the following criteria:

- Are of particularly high value to the organization.
- Would naturally benefit from integration into a single sign-on infrastructure.

- Are indicative of standard programming and deployment platforms within your organization.
- Are generally receptive to the identity management infrastructure.
- Are currently in the early process of deployment and might logically have time lines that coincide with the Access Manager deployment.

You might develop a spreadsheet that can be used to organize the information from the most promising applications. An overall metric can be developed to compare the value of the application to the complexity of its integration. This metric might be considered an application's degree of fitness for deployment. An example of a highly fit application might be a web application that delegates authentication to an application server on which an Access Manager policy agent is installed for security. All user information would be stored in an LDAP directory.

An example of an unfit application might have a text-based interface, running on a mainframe computer. In this case, it would be advantageous to integrate other applications while waiting for a new version of the mainframe application.

The following sections describe types of information that can be gathered when evaluating your organization's applications. This step also helps in determining the resources that will be protected.

Platform Information

General platform information, based on your existing technologies and hardware, can be used to assess the appropriateness of an application as a candidate for integration. Collected platform information might include the following:

- What operating system (including version) do the applications run on?
- Which web containers (including version) do the applications run in?
- What programming model was used to develop the applications ? (such as Java, ASP/.NET, or C)
- Are there plans to upgrade the applications? If so, what is the time line?

LOB applications might also be running third party applications (such as portals, content management databases, or human resource systems). These applications do not always deploy on platforms supported by Access Manager policy agents. If a policy agent is required, determine the deployment criteria of these applications and schedule their deployment based on the availability of a policy agent.

Security Models

It is important to document the existing security models used within the LOB applications. Typically, applications that use external authentication or authorization are candidates for deployment as well as applications that rely on external directory services. Security information might include the following:

- What authentication mechanisms are currently being used?
- Are their special authentication requirements (such as 2-factor authentication)?
- Is there a pluggable interface for external authentication mechanisms?
- What authorization mechanisms are currently being used?
- Can (or should) authorization be externalized?
- What user data repositories are being used? Can these be externalized?
- Who can access the application? Are there existing roles or groups in place? Under what special conditions are they granted access?

Lifecycle of a Session

An identity's session lifecycle is an important topic to consider when evaluating authentication applications. Make sure you have a clear picture of how a user session is created, managed, and destroyed. Clearly document this process because it will be needed during the application's integration.

Customization and Branding

Consider any specific branding or look and feel requirements for the application. Often times, it is important to maintain an individual look and feel or to simply maintain consistency of user experience. Ensure that any customization and branding requirements are noted with your application assessment because time must be scheduled for this activity.

Categorizing Data

Having analyzed your applications and categorized them into fitness levels, you must now begin categorizing the data and services offered by those applications. This information will be used to build a security model. The categorization process itself is a procedure of data and service categorization, followed by cataloguing the existing authentication and authorization systems.

The information collected in Evaluating Applications is used for the former portion of the process. A good methodology might be to organize the collected information into various tiers of security. These tiers would indicate the amount of risk associated with data loss, application compromise, abuse, or other illicit types of access. Using well-defined categories can help to simplify the mapping of resources into a security model incorporating authentication and authorization requirements.

The data or service is separated into four levels of security. The X axis is the data or service and the Y axis is the security level associated with it. Tier 1 is illustrated with a minimal amount of security and might be data applicable to a public web site. Tier 4, on the other hand, requires maximum security and might be financial or human resources (HR) data. Your organization's categorization might have more or less tiers, but this chart shows how typical it is for large amounts of data to have low associated risk, and thus, low security requirements. As risk associated goes up, security requirements also go up. (Usually, there is very little data with high security requirements, and a lot of data with limited or no security requirements.)

The following figure shows the security requirements for data and services within a typical organization.

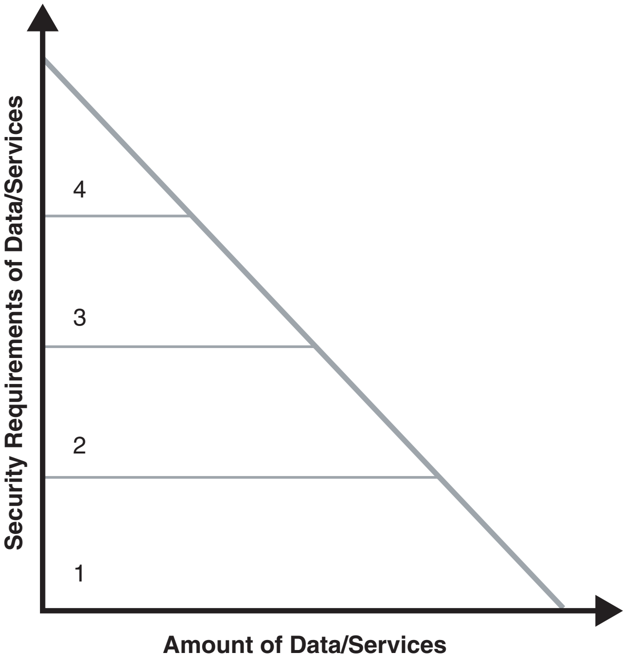


FIGURE 2–1 Security Requirements of Data and Services

Keep in mind that you are planning to build functional groupings of data and service types so that authentication and authorization functions can be mapped to them. Too many tiers can inject extra complexity into your process, while too few tiers might not offer enough flexibility. It is also important to note that there might be data with too much risk to place on the network at all. If relevant, make sure distinctions are made between internal and externally available data. Keep authentication and authorization requirements in mind as you build out these tiers, as well as conditional qualifiers such as such as access time of data and network location.

Mapping To Authentication

With the data categorized according to security level, the next step is to inventory authentication and authorization mechanisms. Using a current list of available authentication mechanisms, associate those mechanisms with the security tiers defined. For example, the following association might be appropriate for the data categorized in the previous figure.

- Tier 1 data might be appropriate for anonymous authentication with no access control.
- Tier 2 data might require password protection only.
- Tier 3 data might require hard token or certificate authentication.
- Tier 4 data might require multi-factor authentication (or might not be placed on the network at all).

You should ensure a clean mapping between authentication requirements and the data and services categorization. If there is none, look for common criteria between those items that do not match. Don't hesitate to make multiple charts if logical distinctions occur.

For example, separate charts can be made for intranet and extranet applications. You might also categorize data based upon a functional security domain such as human resources (HR) or finance. While not a universally applicable tool, categorizing your data in this manner can help you to understand your security requirements and to map them into logically manageable groups.

Mapping To Authorization

Using the data available from your application assessment, examine each of the applications to determine a scalable authorization model. Typically, it is best to look for common groups and roles used across applications. Ideally, these groups and roles will map to functional roles within the organization. You should also determine the source of those groups and roles (where does the membership data live and how is it modeled). For example, the data might be in Sun Java System Directory Server.

If not, custom plug-ins might be required. If a robust grouping model is in place, begin associating each application with existing groups or roles. If not, begin planning a group or role mechanism, finding common relationships between functional user types and access to specific applications. When completed, you should have the following items:

- A clear map of existing groups and roles.
- A clear understanding of where that data lives and who is the authority over its quality and management.

- A clear understanding of new groups or roles that need to be created to facilitate your deployment or to reduce cost and complexity of the deployment.
- A mapping of existing and future grouping mechanisms to your categorized applications.
- Notes on additional conditions required by the applications to allow access to a certain group or role.

With this basic security model (categorization of data, with correlation to authentication and authorization mechanisms), you can now put together a time line to drive your deployment.

Building a Time Line

From the information you have gathered, you should build a preliminary time line. The following sections describes the steps to build a time line for a generic schedule of deployment.

Deployment Design

This phase of the time line is where the concepts, business needs, and user requirements are put in their proper context. A total view of the deployment takes shape. Components are described, technological requirements are defined, and a complete architecture is mapped out. Storyboarding login screens or creating data flow charts are two ways of initiating this design phase.

Proof-of-Concept

A proof-of-concept enables the design to be tested in a business environment. Organizations often have a test case database, a set of pre-configured test cases coupled with their expected results. The proof-of-concept can be applied to this test database, and, if all goes well, the documented results will be equivalent to the new results. A proof-of-concept aims to answer all question posed by the Deployment Design, proving that it meets all needs efficiently and with minimal risk.

It is generally fast allowing for ample time to refine the design based on a limited set of data. There are usually several rounds of proof-of-concept, followed by design refinement. The last round in the proof-of-concept should be integration of some internal applications. Integration of a corporation's shared services often adheres to a standard model of sign-on by early adopters, followed by general participation and, lastly, the stragglers. Demonstrated success with early adopters makes it easier to use those applications as references for general adoption.

Early Adoption

Mission critical or revenue-building applications should not be chosen as your first application. A less risky strategy is to choose an important application that will not completely disrupt business operations if there are issues during roll-out. For example, a divisional portal serves as a natural staging ground for a single sign-on (SSO) roll out, rather than an accounting system at the close of a fiscal period.

Also, limit the number of applications roll-outs in the early phases so process flaws can be driven out, results demonstrated, and immediate success recognized. Minimizing organizational risk while maximizing visibility is the optimal roll-out strategy. This plan positions the deployment team with the appropriate product experience to take on critical applications.

General Participation

Although the deployment project begins with a single application, the requirements of other internal customers should be assessed at the same time so that a general purpose system can be built. The central IT group should be able to accommodate the diverse criteria and schedules of the satellite groups in order to provide service that is representative of the larger organization. Schedules must have sufficiently large windows, allowing the satellite groups time to build changes and upgrades into their application's deployment and quality assurance (QA) cycle.

Production Environment

Following the proof-of-concept, the refined design can be replicated into a production environment. The purpose of a production environment is to demonstrate the function of the design in a non-artificial environment, ensuring its proper behavior. The environment is compared to the behavior as observed in the proof-of-concept, and as defined in the deployment design. It is also tested for stability.

An assessment is made and reports are generated. Early adoption applications go live in the production environment as they are ready. Incrementally phase new applications through the test phase and into production. Other applications are incrementally added to the production environment by working them through the proof-of-concept cycle as the early adopters have been.

Sample time lines are not available, because they vary based upon project complexity. However, this process typically takes place in a span of two to three months.

Deployment Road Map

Mapping out your Access Manager integration is imperative to ensuring its success. This process include collecting information concerning hardware, currently deployed applications, identity data, and access hierarchy. Access Manager deployment can be broken down into the following phases:

1. Identify business objectives such as:

   - Increase operational efficiency.
   - Assure data security.
   - Assure continued productivity by understanding the scope and relationships within the organization and analyzing the behavioral changes needed to support the business objectives.
2. Develop a high-level technology analysis and map it to the business objectives by listing technology services and tools needed to meet business objectives.
3. Define initiatives for each technology service such as:
   - Storing employee history and data accumulated through personalization.
   - Accomplishing password synchronization and identity administration through identity management.
   - Realizing enterprise security through the development of role strategies.
4. Prioritize initiatives based on items such as statistical accuracy, predictability, scope, cost, impact, complexity, behavior, infrastructure, benefit, support, and dependencies.

CHAPTER 3

Technical Requirements

During the technical requirements phase of the solution life cycle you perform a usage analysis, identify use cases, and determine quality of service requirements for the proposed deployment solution. This chapter provides a high-level technical overview of the requirements related to this process for Sun Java™ System Access Manager 7.1, including:

- “Deployment Options” on page 39
- “Hardware Requirements” on page 41
- “Software Requirements” on page 42
- “Web Browser Requirements” on page 44
- “JSSE Encryption Requirements” on page 44
- “Administrative Users” on page 45
- “Anonymous Users” on page 50
- “Access Manager Schema” on page 51

Deployment Options

There are several key factors that an organization should consider when planning for an Access Manager deployment. These considerations generally deal with risk assessment and a growth strategy. For example:

- How many users is your deployment expected to support, and what is your projected growth rate?

  It is critical that user growth and system usage are monitored and that this data is compared with the projected data to ensure that the current capacity is capable of handling the projected growth.
- Do you have plans to add additional services that might impact the current design?

  The architecture being developed now is optimized for the current service. Your future plans should also be examined.

In addition, the architecture should provide a foundation for the objectives detailed in the following sections.

Security

Consider the following options when you are planning for a secure internal and external networking environment:

- Server-based firewalls provide an additional layer of security by locking down port-level access to the servers. As with standard firewalls, server-based firewalls lock down incoming and outgoing TCP/IP traffic.
- Minimization refers to removing all unnecessary software and services from the server in order to minimize the opportunity for exploitation of the vulnerabilities of a system.
- A Split-DNS infrastructure has two zones that are created in one domain. One zone is used by an organization's internal network clients, and the other is used by external network clients. This approach is recommended to ensure a higher level of security. The DNS servers can also use load balancers to improved performance.

High Availability

Deployments strive for no single point of failure (SPOF) as well as continuos availability to its users. Different products achieve availability in different ways; for example, clustering or multi-master replication. The desired high availability refers to a system or component that is continuously operational for a specified length of time. It is generally accomplished with multiple host servers that appear to the user as a single highly available system. In a deployment that meets the minimal requirements (all applications on a single server), the SPOFs might include:

- Access manager web container
- Directory Server
- Java Virtual Machine (JVM)
- Directory Server hard disk
- Access Manager hard disk
- Policy agents

Planning for high availability centers around backup and failover processing as well as data storage and access. For storage, a redundant array of independent disks (RAID) is one approach. For any system to be highly available, the parts of the system should be well-designed and thoroughly tested before they are used. For example, a new application program that has not been thoroughly tested is likely to become a frequent point-of-breakdown in a production system.

Clustering

Clustering is the use of multiple computers to form a single, highly available system. Clustering is often crucial for the Sun Java System Directory Server data store. For example, a clustered multi-master replication (MMR) server pair can increase the availability of each master instance by ensuring availability.

Scalability

Horizontal scaling is achieved by connecting multiple host servers so they work as one unit. A load balanced service is considered horizontally scaled because it increases the speed and availability of the service. Vertical scaling, on the other hand, is increasing the capacity of existing hardware by adding resources within a single host server. The types of resources that can be scaled include CPUs, memory, and storage. Horizontal scaling and vertical scaling are not mutually exclusive; they can work together for a deployment solution. Typically, servers in an environment are not installed at full capacity, so vertical scaling is used to improve performance. When a server approaches full capacity, horizontal scaling can be used to distribute the load among other servers.

Hardware Requirements

The minimum configuration for an Access Manager deployment is a single host server running Access Manager and a web container such as Sun Java System Web Server. Directory Server can be running on the same server or on a different server. In a multiple server deployment, Access Manager instances and their respective web containers are installed on a different host servers, with a load balancer distributing client requests to the various Access Manager instances. Usually, Directory Server and Access Manager are installed on different servers.

For optimum performance, run Access Manager on a 100 Mbytes or greater Ethernet network. A minimum configuration Access Manager deployment (a single server running Access Manager and a web container) should have one or more CPUs, with greatly diminishing returns on processor performance after four CPUs. Two to four CPUs per host server are recommended. A minimum of 512 Mbytes of RAM is necessary for basic testing of the software.

For an actual deployment, 1 Gbytes of RAM is recommended for threads, the Access Manager SDK, the HTTP server, and other internals; 2 Gbytes for basic operation and object allocation space, and 100 Mbytes per 10,000 concurrent sessions. Each Access Manager is recommended to cap out at 100,000 concurrent sessions, after which horizontal load balancing should be used (assuming the 4 Gbytes memory limitation of 32–bit applications).

Software Requirements

Access Manager has specific minimum software requirements, including:

- "Operating System Requirements" on page 42
- "Web Container Requirements" on page 42
- "Directory Server Requirements" on page 43
- "Java Development Kit (JDK) Software Requirements" on page 43
- "Access Manager Session Failover Requirements" on page 43

For the latest information about the software requirements, including supported releases, any required patches, and known limitations, see the *Sun Java System Access Manager 7.1 Release Notes*.

Operating System Requirements

Access Manager 7.1 is supported on these platforms:

- Solaris™ 10 OS on SPARC®, x86, and x64 based systems
- Solaris 9 OS on SPARC and x86 systems
- Red Hat™ Linux OS
- Microsoft Windows OS
- HP-UX OS

For the specific versions of each operating system that are supported, see the *Sun Java System Access Manager 7.1 Release Notes*.

For information about downloading OS patches and patch clusters, see SunSolve Online at `http://sunsolve.sun.com/`.

To list the patches currently installed on a Solaris system, use the `showrev -p` command.

Web Container Requirements

Access Manager 7.1 supports the following web containers for either a full installation or an SDK-only installation:

- Sun Java System Web Server
- Sun Java System Application Server
- BEA WebLogic
- IBM WebSphere Application Server

Note – BEA WebLogic and IBM WebSphere Application Server are not supported on HP-UX systems.

For the supported versions of these web containers, see the *Sun Java System Access Manager 7.1 Release Notes*.

When a policy agent is installed on an Access Manager web container, it uses approximately 10 Mbytes of disk space. This additional space must be considered when configuring the web container. For more information, see the *Sun Java System Access Manager Policy Agent 2.2 User's Guide*.

Directory Server Requirements

Access Manager 7.1 has the following requirements for an LDAP directory server:

- The Access Manager information tree, which contains the following information, is stored in Sun Java System Directory Server:
  - How users authenticate
  - Which resources users can access
  - What information is available to applications after users are given access to resources
- The Access Manager identity repository is used to store user data such as users and groups. Access Manager 7.1 can use Sun Java System Directory Server or an LDAP version 3 (LDAP v3) compliant directory server as the identity repository.

For more information about the Access Manager information tree and identity repository, see the *Sun Java System Access Manager 7.1 Technical Overview*.

Java Development Kit (JDK) Software Requirements

For the specific version of the JDK software required by Access Manager 7.1 , see the *Sun Java System Access Manager 7.1 Release Notes*.

Access Manager Session Failover Requirements

If you are planning to implement Access Manager session failover, these components are required:

- Web container to run Access Manager: Sun Java System Web Server, Sun Java System Application Server, IBM WebSphere Application Server, or BEA WebLogic.
- Sun Java System Directory Server. All Access Manager instances must access the same Directory Server.

- Sun Java System Message Queue. The Message Queue broker cluster manages the session messages between Access Manager instances and the session store database.
- Berkeley DB (`http://www.oracle.com/database/berkeley-db.html`) is the default session store database. Use the version that is distributed with the Sun Java Enterprise System 5 release.

Access Manager session failover is supported on the following platforms:

- Solaris 10 OS on SPARC, x86, and x64 based systems
- Solaris 9 OS on SPARC and x86 systems
- Red Hat Linux OS
- Microsoft Windows OS
- HP-UX OS

For the latest information about the supported versions of these platforms and components, see the *Sun Java System Access Manager 7.1 Release Notes*.

For more information, see Chapter 6, "Implementing Session Failover," in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

Web Browser Requirements

Access Manager administrators and end users use web browsers to perform administrative and user management tasks. For information about the supported web browsers for this release, see the *Sun Java System Access Manager 7.1 Release Notes*.

To access the Access Manager Console, JavaScript must be enabled for the browser.

JSSE Encryption Requirements

To implement secure Internet communications, the following deployment scenarios or activities require Access Manager 7.1 to use Java Secure Socket Extension (JSSE) encryption instead of JSS encryption:

- Client SDK deployment on an SSL-enabled web container instance
- Distributed Authentication deployment on an SSL-enabled web container instance
- Single Access Manager 7.1 WAR file deployment on an SSL-enabled web container instance
- Use of the `com.sun.identity.idm` API with an SSL-enabled Access Manager server
- Access Manager deployment on an SSL-enabled third-party web container instance (BEA WebLogic or IBM WebSphere Application Server)

For JSSE information and downloads, see the following Sun Developer Network (SDN) web site: `http://java.sun.com/products/jsse/`

Administrative Users

When assessing the technical requirements for your Access Manager deployment, consider the following administrative users and accounts:

- "Access Manager Administrative Roles" on page 45
- "Access Manager Administrative Accounts" on page 47
- "Policy Agent Administrative Users" on page 49

Access Manager Administrative Roles

- "Realm Mode Administrative Roles" on page 45
- "Legacy Mode Administrative Roles" on page 46

Realm Mode Administrative Roles

In Access Manager Realm mode, the Delegation plug-in works with the Identity Repository plug-in to determine a network administrator's scope of privileges. Default administrator roles are defined in the Identity Repository plug-in. The Delegation plug-in forms rules that describe the scope of privileges for each network administrator, and also specifies the roles to which the rules apply. The following table lists the roles defined in the Identity Repository and the default rule the Delegation plug-in applies to each role.

TABLE 3–1 Access Manager Roles and Scope of Privileges in Realm Mode

Identity Repository Role	Delegation Rule
Realm Administator	Can access all data in all realms of the Access Manager information tree.
Subrealm Administrator	Can access all data within a specific realm of the Access Manager information tree.
Policy Administrator	Can access all policies in all realms of the Access Manager information tree.
Policy Realm Administrator	Can access policies only within the specific realm of the Access Manager information tree.

The Authentication service and Policy service use the aggregated data to perform the authentication and authorization processes. The code for the Delegation plug-in and Identity Repository plug-in are not public in Access Manager.

Legacy Mode Administrative Roles

In Access Manager Legacy mode, delegated administration of the LDAP entries (mapped to each identity-related object in Access Manager) are implemented through the use of pre-defined roles and access control instructions (ACIs). Default administrative roles and their defined ACIs are created during Access Manager installation and can be viewed and managed using the Access Manager Console. In Access Manager 7.1 in Realm mode, roles depend on policies rather then ACIs.

When an Access Manager identity-related object is created, the appropriate administrative roles (and thus, corresponding ACIs) are also created and assigned to the LDAP entry for that object. The role can then be assigned to an individual user who maintains control of that object's node. For example, when Access Manager creates a new organization, several roles are automatically created for it and stored in Directory Server:

- Organization Administrator has read and write access to all entries in the configured organization.
- Organization Help Desk Administrator has read access to all entries in the configured organization and write access to the `userPassword` attribute in those entries.
- Organization Policy Administrator has read and write access to all policies in the organization.

The assignation of any of these roles to a user gives that user all the permissions accorded that role.

The following table summarizes the Access Manager administrator roles and the permissions that apply to each one.

TABLE 3–2 Default and Dynamic Roles and Their Permissions in Legacy Mode

Role	Administrative Suffix	Permissions
Top-level Organization Admin (amadmin)	Root level	Read and write access to all entries (such as roles, policy, and groups) under top-level organization.
Top-level Organization Help Desk Admin	Root level	Read and write access to all passwords under top-level organization.
Top-level Organization Policy Admin	Root level	Read and write access to policies at all levels. Used by sub-organizations to delegate referral policy creation.
Organization Admin	Organization only	Read and write access to all entries (such as roles, policy, and groups) under the created sub-organization only.

TABLE 3–2 Default and Dynamic Roles and Their Permissions in Legacy Mode *(Continued)*

Role	Administrative Suffix	Permissions
Organization Help Desk Admin	Organization only	Read and write access to all passwords under the created sub-organization only.
Organization Policy Admin	Organization only	Read and write access to all policies under the created sub-organization only.
Container Admin	Container only	Read and write access to all entries (such as roles, policy, and groups) under the created container only.
Container Help Desk Admin	Container only	Read and write access to all passwords under the created container only.
Group Admin	Group only	Read and write access to all entries (such as roles, policy, and groups) under the created group only.
People Container Admin	People Container only	Read and write access to all entries (such as roles, policy, and groups) under the created people container only.
User (self-administrator)	User only	Read and write access to attributes in the user entry only (except for user attributes such as `nsroledn` and `inetuserstatus`).

Using roles instead of group-based ACIs is more efficient and requires less maintenance. Filtered roles are simpler for LDAP clients, because they can just ask for the `nsRole` attribute of a user. Roles do suffer though from scope limitations, where a role must be a peer of a parent of a member of that role.

For more information about default ACIs, see the Access Manager Console Online Help.

Access Manager Administrative Accounts

During the installation of Access Manager, the following administrative accounts are created:

- Administrator user ID (`amadmin`) is the Access Manager top-level administrator that has unlimited access to all entries managed by Access Manager. You cannot change the default name, `amadmin`.

  During installation, you must provide a password for `amadmin`. To change the `amadmin` password after installation, use the Access Manager Administration Console.
- Bind DN user for LDAP, Membership, and Policy services (`amldapuser`) is the administrative user that has read and search access to all Directory Server entries. You cannot change the default name, `amldapuser`.

During installation, you must provide a password for amldapuser. Do not use the same password that you used for amadmin. To change the amldapuser password after installation, use the Directory Server Console or the ldapmodify utility.

If you change the amldapuser password, you must also modify the LDAP authentication service and policy configuration services to reflect the change (amldapuser is the default user used in these services). You must make changes in each organization where these services are registered.

- Proxy user (puser) can take on any user's privileges (for example, an organization administrator or end user).
- Admin user (dsameuser) is used for binding purposes when the Access Manager SDK performs operations on Directory Server that are not linked to a particular user (for example, retrieving service configuration information).

Both puser and dsameuser have an associated password that is stored in encrypted format in the serverconfig.xml file, in the following directories:

- Solaris systems: /etc/opt/SUNWam/config
- Linux and HP-UX systems: /etc/opt/sun/identity/config
- Windows systems: *javaes-install-dir*\identity\config

  The *javaes-install-dir* variable represents the Java ES 5 installation directory. The default value is C:\Program Files\Sun\JavaES5.

After installation, it is recommended that you change the password for puser and dsameuser, but do not use the same password that you used for amadmin or amldapuser. To change the puser or dsameuser password, use the ampassword utility:

- The ampassword --admin (or -a) option changes the password for dsameuser. (This option does not change the amadmin password.)
- The ampassword --proxy (or -p) option changes the password for puser.

Changing the puser or dsameuser password depends on your deployment.

If Access Manager is deployed on a single host server:

1. Use the ampassword utility to change the respective password in Directory Server and in the local serverconfig.xml file.
2. Restart the Access Manager web container.

If Access Manager is deployed on multiple host servers:

1. On the first server, use the ampassword utility to change the respective password in Directory Server and in the local serverconfig.xml file.
2. Encrypt the new password using the ampassword --encrypt (or -e) option.
3. On each additional server where Access Manager is deployed, change the password manually in the serverconfig.xml file, using the new encrypted password from Step 2.

4. On each server where you changed the password, including the first server, restart the Access Manager web container.

For information about the ampassword utility, see the *Sun Java System Access Manager 7.1 Administration Reference*.

Policy Agent Administrative Users

A Policy Agent in the Access Manager Policy Agent 2.2 software set authenticates to Access Manager using a user name and password stored in its AMAgent.properties file. The process is similar but slightly different for “Web Agents” on page 49 and “J2EE Agents” on page 49.

Web Agents

A Web Agent uses the following properties in the AMAgent.properties file to store its user name and password used to authenticate to Access Manager:

- com.sun.am.policy.am.username contains the name of the user that the Web Agent uses to login to Access Manager. The default name is UrlAccessAgent.
- com.sun.am.policy.am.password contains the encrypted password of the user that the Web Agent uses to login to Access Manager. The password must be encrypted using the crypt_util utility.

When an Access Manager instance is initially configured, the Java ES installer or the amconfig script creates the amService-UrlAccessAgent user in the top-level realm with the same password as the amldapuser user.

By default, all Web Agents use the same user name and password to authenticate to an Access Manager instance. To improve security and to allow each Web Agent to use a unique name and password, you can create an agent profile in the Access Manager Administration Console for a Web Agent to use for authentication. For more information, see “Using an Agent Profile for Authentication” on page 50.

J2EE Agents

A J2EE Agent communicates with Access Manager with a user name (agent ID) and password through an agent profile created in the Access Manager Administration Console. After the agent profile is created, the J2EE Agent then uses the following properties in its AMAgent.properties file to store the user name (agent ID) and password:

- com.sun.identity.agents.app.username contains the user name (agent ID) that the J2EE Agent uses to login to Access Manager.
- com.iplanet.am.service.secret contains the encrypted password of the user name (agent ID) that the J2EE Agent uses to login to Access Manager. The password must be encrypted using the agentadmin utility with the --encrypt option.

See the next section for information about agent profiles.

Using an Agent Profile for Authentication

To authenticate to Access Manager, a J2EE Agent requires that you create an agent profile in the Access Manager Administration Console. A Web Agent can also use an agent profile, which allows each Web Agent to have a unique user name (agent ID) and password. For the steps to create an agent profile, see the Access Manager Console online Help.

An agent profile also allows you to change the password and user name (agent ID) for a Policy Agent, as required by your deployment. To change a password and user name (if required), follow these general steps:

1. Log in to the Access Manager Console as the Access Manager administrator (`amadmin`).
2. In the agent profile for the Policy Agent, change the password and user name (agent ID), if required. Save the profile.
3. Encrypt the new agent password from Step 2 using the `crypt_util` utility for Web Agents or the `agentadmin` utility with the `--encrypt` option for J2EE Agents.
4. Set the following properties in the Policy Agent's `AMAgent.properties` file:
   - For Web Agents: Set the `com.sun.am.policy.am.password` property to the new encrypted password from Step 3. If you also changed the user name (agent ID), set the `com.sun.am.policy.am.username` property to the new user name (agent ID) from Step 2.
   - For J2EE Agents: Set the `com.iplanet.am.service.secret` property to the new encrypted password from Step 3. If you also changed the user name (agent ID), set the `com.sun.identity.agents.app.username` property to the new user name (agent ID) from Step 2.
5. Restart the Web Agent web container for the new password (and user name if you changed it) to take effect.

For more detailed information about creating and configuring agent profiles and encrypting passwords, see the Access Manager Policy Agent 2.2 documentation collection:

http://docs.sun.com/coll/1322.1

Anonymous Users

If the Anonymous module is enabled, anonymous users can log into Access Manager without providing a password. The default anonymous user is `anonymous`, but you can change this name or define a list of anonymous users by setting the following realm attributes in the Access Manager Administration Console:

- Valid Anonymous Users specifies a list of user IDs that you want to have anonymous access.

- Default Anonymous User Name allows you to specify a name other than the default value of anonymous. This name is used if the `Valid Anonymous User` list is empty.
- Case Sensitive User IDs specifies that anonymous user IDs must case-sensitive. By default, user IDs are not case-sensitive.
- Authentication Level limits anonymous users to specific types of access, such as read and search only. The default value is 0.

These attributes apply to a realm, so you can set different anonymous access attributes to specific realms.

For information about enabling the Anonymous module and creating anonymous users, see the Access Manager Console online Help.

Access Manager Schema

In general, a schema is a set of rules imposed on data that defines how it is stored. Sun Java System Directory Server uses the Lightweight Directory Access Protocol (LDAP) schema to define how its data is stored. Object classes define attributes in a LDAP schema. In Directory Server, each data entry must have one or more object class(es) to specify the type of object the entry describes and define the set of attributes it contains. Each entry then is basically a set of attributes and their corresponding values and the list of object classes to which these attributes correspond.

Access Manager uses Sun Java System Directory Server as its data repository, which includes the Access Manager schema that extends the Directory Server schema. When Access Manager is installed, the Access Manager schema, from the `ds_remote_schema.ldif` and `sunone_schema2.ldif` files, is integrated with the Directory Server schema. The `ds_remote_schema.ldif` file defines the LDAP object classes and attributes that are specifically used by Access Manager. The `sunone_schema2.ldif` file loads the Access Manager LDAP schema object classes and attributes.

You can view the `ds_remote_schema.ldif`, `sunone_schema2.ldif`, and other Access Manager LDIF files in the following directories:

- Solaris systems: `/etc/opt/SUNWam/config/ldif`
- Linux and HP-UX systems: `/etc/opt/sun/identity/config/ldif`
- Windows systems: *javaes-install-dir*`\identity\config\ldif`

  The *javaes-install-dir* variable represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

Marker Object Classes

Identity entries created using the Access Manager console and stored in Directory Server are appended with marker object classes. Marker object classes define the designated entries as those which Access Manager will manage. The object classes will not interfere with other aspects of the directory tree, such as servers or hardware. As well, existing identity entries can not be managed using Access Manager until they are modified to include these object classes. More detailed information about the marker object classes can be found in the Access Manager Developer's Guide. For information about migrating existing Directory Server data for use with Access Manager, see the *Sun Java Access Manager 6 2005Q1 Migration Guide*.

Schema Limitations

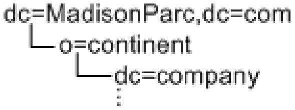
Access Manager abstractly represents the entries it manages. This means, for example, that an organization in Access Manager is not necessarily the same as an organization in Directory Server. Whether a specific Directory Information Tree (DIT) can be managed or not depends on how the you choose to represent or manage your directory entries and whether your DIT fits into the limitations of each Access Manager type.

The following sections describe these limitations of the Access Manager schema. At the end of this section, several "Examples of Unsupported DITs" on page 54 are included.

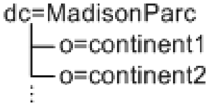
Only One Type of Entry Can be Marked as an Organization

By adding the Access Manager `sunISManagedOrganization` auxiliary class to any entry, Access Manager can manage this entry as if it is an organization. However, only one type of entry may be marked as an organization in Access Manager. For example, if you have an entry `o=sun` and another entry `dc=ibm` in your DIT, you cannot mark them both as organizations.

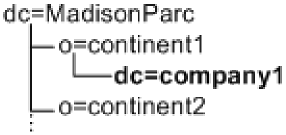
In the following example, if you want both the `dc` and `o` entries to be organizations, the DIT structure will not be manageable using Access Manager:



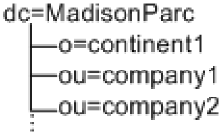
The entry at the Access Manager root suffix, however, does not count as one entry. Therefore, in the following example, the DIT structure can be managed by Access Manager:



If you were able to add dc=company1 below o=continent1, then this DIT would be manageable only if dc is marked as a container. Container is another abstract type in Access Manager that typically maps to an OrganizationalUnit. In most DITs, you would add the iplanet-am-managed-container entry to all OrganizationlUnits.



However, you could add this marker object class to any entry type. The DIT structure in the following example is allowed:



In this example, because you cannot mark both o= and ou= entries as organizations, you could mark the o= entries as organization and the ou= entries as containers. When exposed in the console, both organizations and containers have the same options. You can create subordination or subcontinents, people containers, groups, roles, and users under both of them.

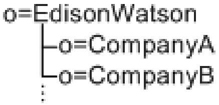
People Containers Must be Parent Entries for Users

Another abstract entry type is the people container. The Access Manager type assumes that this entry is a parent entry for users. When you mark an entry as a people container with iplanet-am-managed-people-container, the UI will assume it can only contain sub-people containers or users. The attribute OrganizationUnit is typically used as a people container, but any entry can be this type in Access Manager as long as it has the iplanet-am-managed-people-container object class and it has a Access Manager manageable parent of type organization or container.

Only One Organization Description is Allowed in the Access Manager XML

The Access Manager organization is defined in amEntrySpecific.xml. Only one organization description is allowed in this file. As a result, when you customize directory entry properties, or create administration pages or search pages in the UI, your custom attributes apply globally to the entire Access Manager configuration. This Access Manager requirement may not meet the needs of some companies, especially hosting companies, that require different display attributes for each organization in the deployment.

In the following example, Edison-Watson is a hosting company that provides internet services to a number of companies. CompanyA wants to display fields for capturing a user's name First Name, Surname, and Badge Number. CompanyB wants to display fields for capturing a user's First Name, Last Name, and Employee Number.

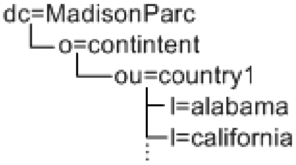


The organization description is defined at the root level (`o=EdisonWatson`), and not at the organization level. By default, the UI for both CompanyA and CompanyB must be identical. Also, all services globally define attributes to be of the subschema type user. So if CompanyA has attributes for its users in the auxiliary class `CompanyA-user`, and CompanyB has attributes in `CompanyB-user` then CompanyB's attributes will be overridden and will not be displayed.

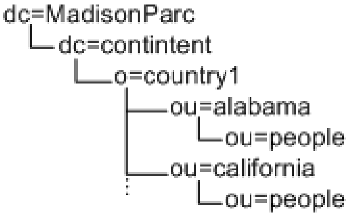
As a workaround, you can modify the ACIs to work for user display. However, this workaround will not address the attributes in Search and Create windows.

Examples of Unsupported DITs

In the following example, you would need three types of organization makers: `o`, `ou`, and `l`. Assuming that `l=california` and `l=alabama` are not a people containers, this DIT would not work with Access Manager:



In the following example, you would need three types of Access Manager markers (`dc`,`o`,`ou`) plus the people container type (`ou=people`). Under these assumptions, the DIT would not work with Access Manager:



CHAPTER 4

Logical Design with Access Manager

During the logical design phase of the solution life cycle, you design a logical architecture showing the interrelationships of the logical components of the solution. The logical architecture and the usage analysis from the technical requirements phase form a deployment scenario, which is the input to the deployment design phase. This chapter contains the following sections about logical design for Sun Java™ System Access Manager:

- "About Logical Architectures" on page 55
- "Access Manager Components" on page 56
- "Java ES Components That Use Access Manager" on page 57
- "Example Access Manager Logical Architectures" on page 58

About Logical Architectures

A logical architecture identifies the software components needed to implement a solution, showing the interrelationships among the components. The logical architecture and the quality of service requirements determined during the technical requirements phase form a deployment scenario. The deployment scenario is the basis for designing the deployment architecture, which occurs in the next phase, deployment design.

Designing a Logical Architecture

When you design a logical architecture, use the use cases identified during the technical requirements phase to determine the Java Enterprise System (Java ES) components that provide the services necessary for the solution. You must also identify any components providing services to the components you initially identify.

You place the Java ES components within the context of a multi-tiered architecture according to the type of services that they provide. Understanding the components as part of a multi-tiered architecture helps you later determine how to distribute the services provided by the components and also helps determine a strategy for implementing quality of service (such as scalability, availability, and others.)

For more detailed information about logical architectures and the solution life cycle, see the *Sun Java Enterprise System Deployment Planning Guide* in the following documentation collection: `http://docs.sun.com/coll/1286.2`.

Access Manager Components

An Access Manager deployment includes the following products and components:

- "Web Container" on page 56
- "Directory Server" on page 56
- "Message Queue and Berkeley DB for Session Failover" on page 57

Web Container

Access Manager must run in one of the following web containers:

- Sun Java System Web Server
- Sun Java System Application Server
- BEA WebLogic Server
- IBM WebSphere Application Server

For the specific versions of each web container that are supported, see the *Sun Java System Access Manager 7.1 Release Notes*.

Directory Server

Access Manager requires an LDAP directory server for these entities:

- "Access Manager Information Tree" on page 57
- "Identity Repository" on page 57

Access Manager Information Tree

Access Manager 7.1 requires Sun Java System Directory Server to store the Access Manager information tree. Access Manager creates and maintains the Access Manager information tree, which includes the following information pertinent to system access:

- How users authenticate
- Which resources users can access
- What information is available to applications after users are given access to resources

Identity Repository

Access Manager requires an identity repository to store user data such as users and groups. Previous versions of Access Manager required Sun Java System Directory Server as the identity repository. However, in addition to Sun Java System Directory Server, Access Manager 7.1 also supports an LDAP version 3 (LDAP v3) compliant directory server.

Message Queue and Berkeley DB for Session Failover

If you are planning to implement session failover, Access Manager requires these additional components:

- Sun Java System Message Queue. The Message Queue broker cluster manages the session messages between Access Manager instances and the session store database.
- Berkeley DB (`http://www.oracle.com/database/berkeley-db.html`) is the default session store database.

Java ES Components That Use Access Manager

Access Manager is usually deployed with other Java ES component products, including:

- Sun Java System Portal Server, Sun Java System Messaging Server, Sun Java System Calendar Server, Sun Java System Instant Messaging, and Sun Java System Communications Express: To provide single sign-on (SSO)
- Sun Java System Web Server: To provide optional access control.

Example Access Manager Logical Architectures

This section provides the following scenarios as examples of logical architectures for Access Manager solutions, including:

- "Access Manager Web Deployment" on page 58
- "Access Manager Multiple Server Deployment" on page 59
- "Java Application Deployment" on page 60
- "Access Manager Session Failover Deployment" on page 61
- "Access Manager and Portal Server Deployment" on page 64
- "Federation Management, SAML, and Web Services" on page 65

Access Manager Web Deployment

A common Access Manager deployment has a web browser accessing an application or resource deployed on a web server. The application or resource is protected by Access Manager and communicates with it using a policy agent also installed on the web server. The web server might also have the Access Manager SDK deployed. This scenario does not restrict the number of web servers on a machine or the instances of Access Manager deployed on multiple machines. For example, a machine might have multiple web servers, each deploying an instance of Access Manager. Similarly, multiple web servers might also be running on different machines, each deploying an instance of Access Manager.

The following figure shows an Access Manager web deployment scenario.

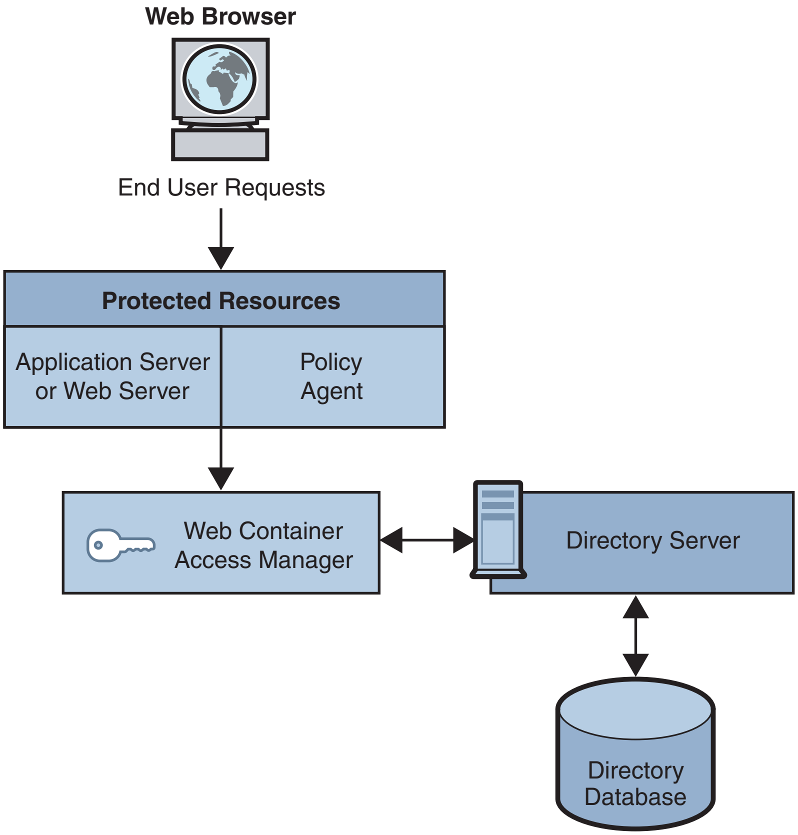


FIGURE 4–1 Access Manager Web Deployment

Access Manager Multiple Server Deployment

An Access Manager multiple server deployment has two or more host servers, with one or more instances of Access manager installed on each host server. Each Access Manager instance accesses the same Directory Server. You can configure the Directory Server instances in a multiple master replication (MMR) configuration, if required for your deployment.

The Access Manager instance installed on the first host server points to an instance of Directory Server. During installation using the Java ES installer, you can choose an existing Directory Server with or without an existing directory information tree (DIT), depending on your deployment.

You install subsequent instances of Access Manager on other host servers by running the Java ES installer, with the Access Manager instance pointing to a Directory Server with an existing DIT. Access Manager then does not write any information to Directory Server because it recognizes the Directory Server as already existing.

The following figure shows multiple Access Manager instances on different host servers with one Directory Server.

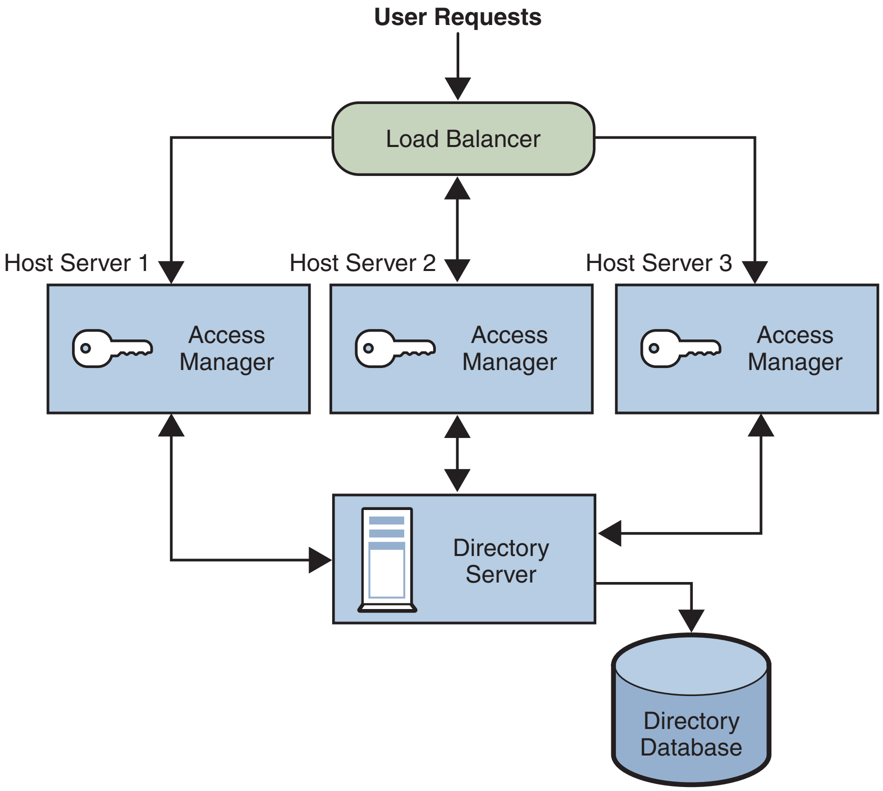


FIGURE 4–2 Multiple Access Manager Instances With One Directory Server

For more information, see Chapter 3, "Deploying Multiple Access Manager Instances," in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

Java Application Deployment

Another common scenario for Access Manager allows Java applications to access an Access Manager SDK installed directly on the server where they are deployed. This scenario requires an additional server with an instance of a web container (such as Sun Java System Web Server or Sun Java System Application Server) running at least one instance of Access Manager. This server also maintains the information to provide single sign-on (SSO). The following figure shows a Java application deployment scenario.

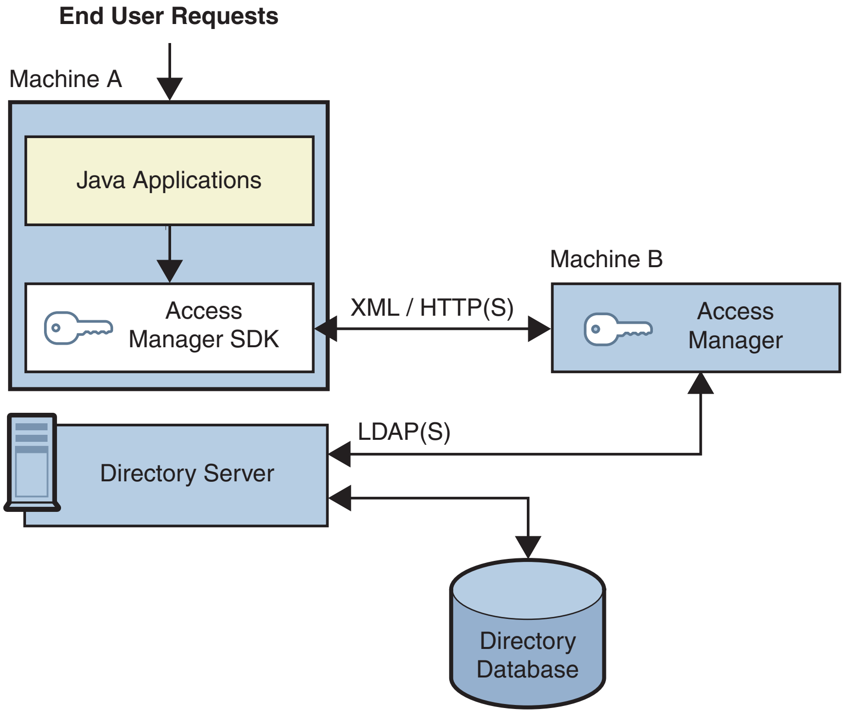


FIGURE 4–3 Java Application Deployment

Access Manager Session Failover Deployment

Access Manager provides a web container independent session failover implementation using Sun Java System Message Queue (Message Queue) as the communications broker and the Berkeley DB as the default session store database. Access Manager session failover retains a user's authenticated session state in the event of a single hardware or software failure, which allows the user's session to fail over to a secondary Access Manager instance without losing any session information or requiring the user to login again.

Overview of Access Manager Session Failover

Access Manager 7.1 session failover includes these components:

- Two or more instances of Access Manager 7.1, with each instance running on a supported web container on two or more host servers.
- Message Queue broker cluster, which manages the session messages between the Access Manager instances and the session store database.
- Berkeley DB (`http://www.oracle.com/database/berkeley-db.html`), as the session store database. The Berkeley DB client daemon is `amsessiondb`.

Access Manager session failover follows the Message Queue publish/subscribe (topic destinations) delivery model:

1. When a user initiates, updates, or ends a session, Access Manager publishes a session creation, update, or deletion message to the Message Queue broker cluster.
2. The Berkeley DB client (`amsessiondb`) subscribes to the Message Queue broker cluster, reads the session messages, and stores the session operations in the database.

If an Access Manager instance fails due to a single hardware or software problem, a user's session associated with that instance fails over to a secondary Access Manager instance, as follows:

1. The secondary Access Manager instance publishes a query request to the Message Queue broker cluster for the user's session information.
2. The Berkeley DB clients (`amsessiondb`) subscribing to the same session request topic on the Message Queue broker cluster receive the query request retrieve the corresponding entry from the session database, and then publish the user's session information to the Message Queue broker cluster with the session response topic.
3. The secondary Access Manager instance subscribing to the session response topic receives the response with the user's session and continues without losing any session information or the user having to login again.

If a Message Queue broker fails, Access Manager continues to operate in non-session failover mode. When the Message Queue broker is later restarted, Access Manager returns to session failover mode.

For more information about the Message Queue components and the publish/subscribe delivery model, see the *Sun Java System Message Queue 3.7 UR1 Technical Overview*.

Session Failover Deployment Scenario

The following figure shows a basic scenario with two host servers, each running an Access Manager instance on a web container, the Message Queue broker cluster, and the Berkeley DB client (`amsessiondb`). The load balancer distributes client requests to the Access Manager instances. Both Access Manager instances access the same Directory Server (not shown in the figure).

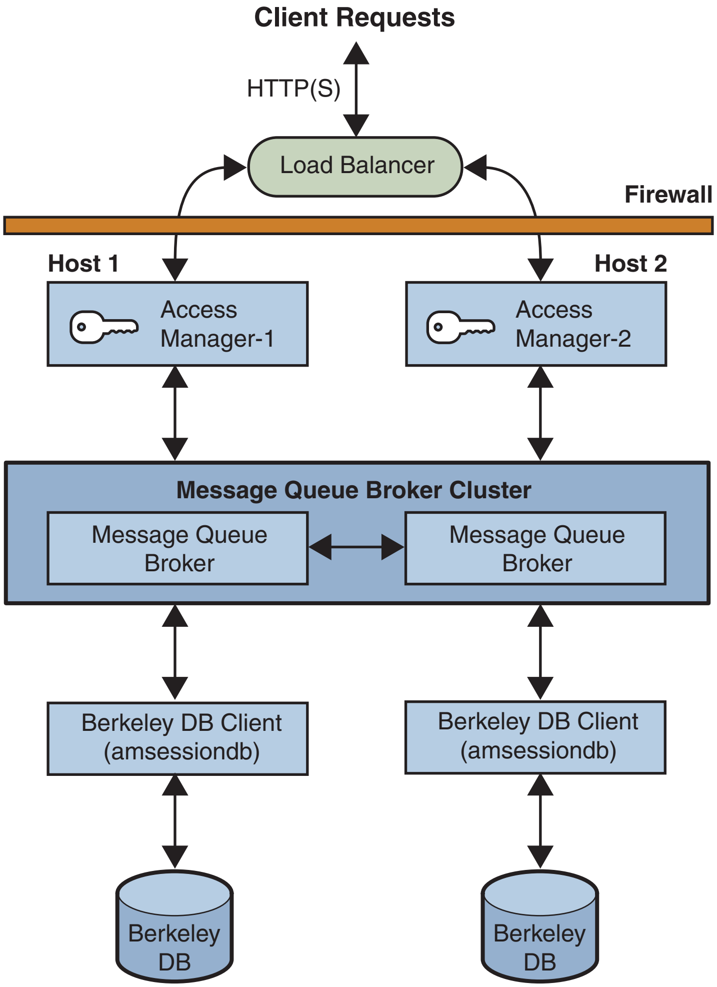


FIGURE 4–4 Access Manager Session Failover Basic Deployment Scenario

You can add additional sites similar to the one shown in the figure, with each site accessing the same Directory Server. Session failover, however, occurs only for the Access Manager instances within a site; cross-site session failover is not supported in the current release.

For more information, see Chapter 6, “Implementing Session Failover,” in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

Access Manager and Portal Server Deployment

For the Java Enterprise System 5 release, you can deploy Access Manager with Portal Server either on the same physical server or on multiple servers.

Installation on a Single Server

In this scenario, Access Manager and Portal Server are installed on the same physical server. You must also install or have access to an installed version of Directory Server, which can be either on the same server or a remote server.

To install these components, run the Java Enterprise System installer in a single session and make these selections:

- On the Component Selection panel, select these products and subcomponents:
  - Under Communication & Collaboration Services, select Portal Server.
  - Under Directory & Identity Services, select Access Manager 7.1 and its subcomponents:
    - Identity Management and Policy Services Core
    - Access Manager Administration Console
    - Common Domain Services for Federation Management
    - Access Manager SDK

      By default, when you select Portal Server, the installer installs only the Access Manager SDK, so you must specifically check the other subcomponents.

  Install and configure one of the following web containers:
  - Sun Java System Application Server
  - Sun Java System Web Server

Installation on Multiple Servers

In this scenario, Portal Server will access Access Manager on a local server from a remote server. You must also install or have access to an installed version of Directory Server, which can be either on a local or remote server:

- On the local server, install Access Manager and a web container. You must install and configure the components on this server before you install and configure the components on the remote server.
- On the remote server, install Portal Server and the Access Manager SDK. You do not need to select the other Access Manager subcomponents on the remote server.

For more information about deploying Access Manager and Portal Server, see the *Sun Java System Portal Server 7.1 Deployment Planning Guide*.

Federation Management, SAML, and Web Services

In 2001, Sun Microsystems joined with other companies to form the Liberty Alliance Project. This project defines standards for developing identity-based infrastructures, software, and web services.

Initially, Access Manager implemented the Liberty Identity Federation Framework (Liberty ID-FF) specification, which comprises a framework for account federation and single sign-on (SSO). Subsequent releases of Access Manager added new features, as defined in version 1.2 of the Liberty ID-FF specifications and the version 1.0 specifications of the Liberty Identity Web Services Framework (Liberty ID-WSF).

The Liberty ID-WSF framework defines a web services stack that you can use to support the Liberty Alliance Project business model. Example services include a personal profile service, discovery service, authentication service, and SOAP binding service. These web services leverage the Liberty ID-FF for principal authentication, federation, and privacy protections.

Access Manager also implements a Security Assertion Markup Language (SAML) service to exchange security information. Both the SAML 1.0 and 1.1 specifications are supported.

For more information, see the *Sun Java System Access Manager 7.1 Federation and SAML Administration Guide*. This guide includes an introduction to the specifications and information about how Access Manager has implemented them. It also includes configuration information, use cases, and summaries of the application programming interface (API).

Sun Java System Federation Manager

Sun Java System Federation Manager 7.0 2005Q4 is a lightweight server application that helps companies to quickly build interoperable identity and authentication services based on the Liberty Alliance Project specifications. These services work with and complement existing or newly deployed federation technologies, such as web access management solutions and authentication authorities.

You can use Federation Manager to build a reusable, standards-based framework to exchange security assertions, user attributes, and policies across a distributed network of partners. Federation Manager is a standalone product that can work with any Liberty or SAML-compliant product. You do not have to install Access Manager in order to use Federation Manager. For more information, see the following documentation collection:

http://docs.sun.com/coll/1321.1

Sun Java System Access Manager Policy Agent 2.2 for Sun Java System Application Server 9.0 / Web Services

The Sun Java System Access Manager Policy Agent 2.2 for Sun Java System Application Server 9.0 / Web Services plugs into Sun Java System Application Server Platform Edition 9.0 to provide message-level security and support for both Liberty Alliance Project token profiles and Web Services-Interoperability Basic Security Profiles (WS-I BSP). This agent provides both an HTTP authentication agent and a SOAP authentication agent and uses Access Manager 7.1 for all authentication decisions.

For more information, including the installation procedure for the agent, see the link to the *Sun Java System Access Manager Policy Agent 2.2 Guide for Sun Java System Application Server 9.0/Web Services*.

Note – Sun Java System Application Server Platform Edition 9.0 is not a Java Enterprise System 5 component. For more information, see the following documentation collection:

`http://docs.sun.com/coll/1343.3`

CHAPTER 5

Deployment Design with Access Manager

During the deployment design phase of the solution life cycle, you design a high-level deployment architecture and a low-level implementation specification, and prepare a series of plans and specifications necessary to implement the solution. Project approval occurs in the deployment design phase. This chapter includes the following sections about deployment design with Sun Java™ System Access Manager:

- "Using a Load Balancer" on page 67
- "Multiple JVM Environment" on page 69
- "Directory Server Replication Considerations" on page 69
- "Directory Server With a Firewall" on page 74

Using a Load Balancer

In most deployments, Access Manager is configured with a load balancer to distribute user requests between two or more Access Manager instances. The load balancer can be implemented with hardware, software, or a combination of both. The following figure shows an Access Manager deployment with a load balancer.

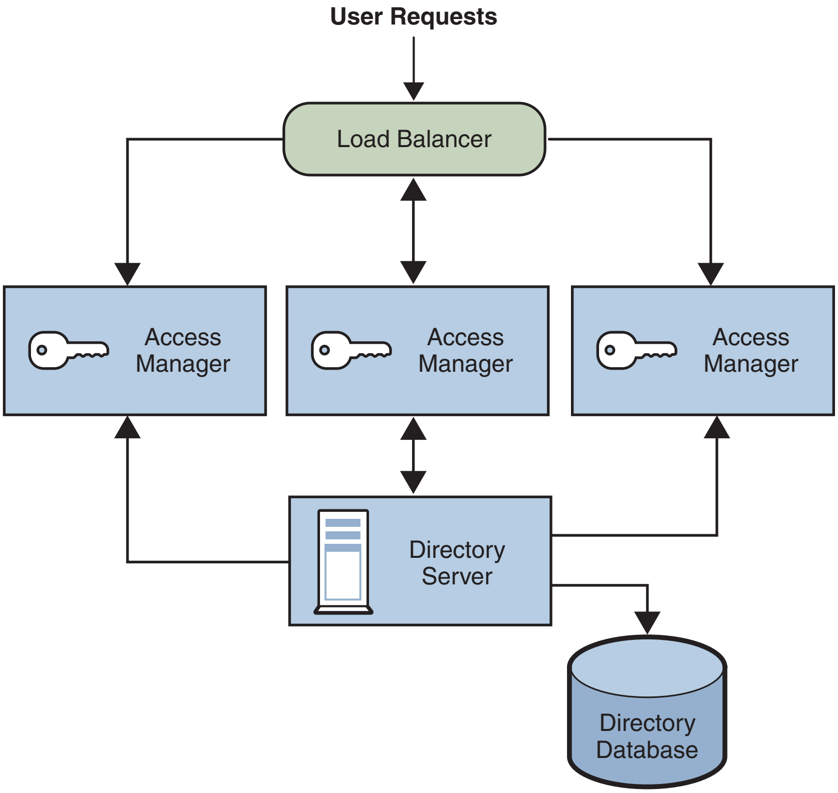


FIGURE 5–1 Access Manager Configuration With a Load Balancer

Cookie-Based Sticky Request Routing

A load balancer deployed with Access Manager must support sticky sessions. A sticky session specifies that once a session is created by a specific Access Manager instance, subsequent requests from the user will continue to be routed to that same instance, in order to preserve session information. Because Access Manager uses cookies to relay session information, the load balancer must redirect the request to the Access Manager instance that created the session.

Therefore, Access Manager has implemented cookie-based sticky request routing, which prevents a client request from being misrouted to an incorrect Access Manager server (that is, to a server that is not hosting the session). This feature prevents the need for back-channel communication between servers and thus improves Access Manager performance.

For more information, see "Configuring Cookie-Based Sticky Request Routing" in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

Multiple JVM Environment

Access Manager services are supported in multiple Java Virtual Machine (JVM) environments. That is, an instance of Sun Java System Application Server can be configured to have multiple JVMs with Access Manager services running in all of them. The Access Manager architecture imposes no restrictions on the deployment with regards to the number of Sun Java System Application Server instances within a machine, the number of Access Manager services across multiple machines, or the number of JVMs that a single Application Server can have.

For more information about the multiple JVM environment, see the Sun Java System Application Server documentation: http://docs.sun.com/coll/1310.3.

Directory Server Replication Considerations

Two methods to improve Access Manager performance and response time are using load balancing across replicated Directory Servers and locating replicated servers closer to users. Directory Server can be set up in single-supplier or multi-supplier configurations. Load-balancing applications such as Sun Java System Directory Proxy Server can also be used. Directory Proxy Server dynamically performs proportional load balancing of LDAP operations across a set of configured Directory Servers. If one or more Directory Server instances become unavailable, the load is proportionally redistributed among the remaining servers. When the server comes back on line, the load is proportionally and dynamically reallocated.

Directory Server replication must be configured before installing Access Manager. This configuration ensures that the supplier and consumer databases are synchronized correctly, allowing time to verify that referrals and updates are synchronized properly.

When Access Manager is installed for replication purposes, each instance of Directory Server and each instance of Access Manager, must be configured with the same values for the following:

- Directory Manager
- Directory Manager Password
- Directory Server Administrator ID
- Server Administrator Password
- Base suffix
- Default organization

Configuring For Replication

Access Manager can be configured to work with single-supplier or multiple-supplier replication. The following figure shows a single-supplier configuration where the consumer is a read-only database. Requests for write operations are referred to the supplier database. This configuration provides some measure of enhanced server performance by distributing the workload to more than one directory.

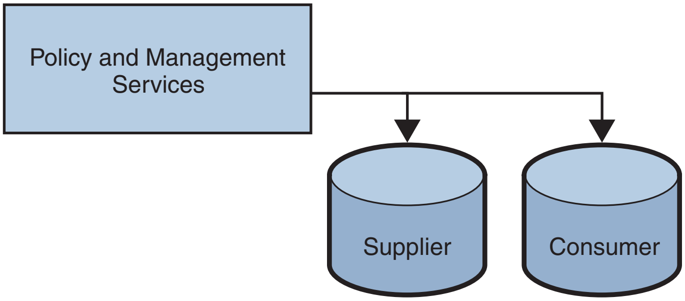


FIGURE 5–2 Single-Supplier Directory Server Replication

The following figure shows a multiple-supplier configuration, or multi-master replication (MMR), using multiple instances of Access Manager. This configuration provides failover protection as well as high availability, resulting in further enhanced server performance.

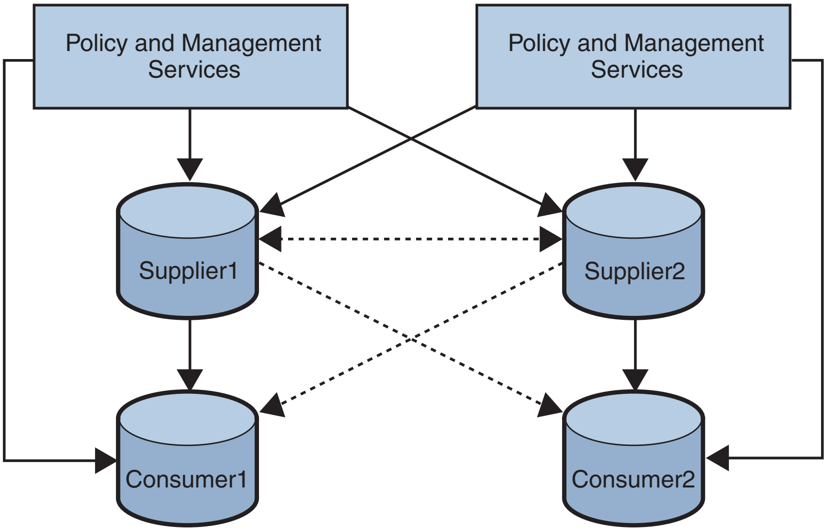

Policy and Management Services
Policy and Management Services
Supplier1
Supplier2
Consumer1
Consumer2

FIGURE 5–3 Multiple-Supplier Directory Server Configuration

Follow these steps to configure replication at the root or top level of the Access Manager directory tree when Access Manager has not yet been installed or to configure replication at the default organization level:

1. Install the supplier and consumer Directory Server instances.

   See the *Sun Java Enterprise System 5 Installation Guide for UNIX* for detailed instructions.

2. Set up replication agreements between the supplier and consumer and verify that the directory referrals and updates are working properly.

   You might need to migrate existing Directory Server data to work with this version of Access Manager. For information, see the *Sun Java System Access Manager 6 2005Q1 Migration Guide*.

3. If you are deploying Access Manager and Directory Server for the first time, or if there is no plan to use existing user data, run the Java ES installation program to install Access Manager.

   During installation, answer yes when asked if there is an existing Directory Server, and specify the host name and port number for a supplier Directory Server you installed in "Configuring For Replication" on page 69.

4. On the host server where Access Manager is installed, modify the `AMConfig.properties` file in the following directory, depending on your platform:

   - Solaris systems: `/etc/opt/SUNWam/config`
   - Linux and HP-UX systems: `/etc/opt/sun/identity/config`
   - Windows systems: *javaes-install-dir*`\identity\config`

     The *javaes-install-dir* variable represents the Java ES 5 installation directory. The default value is `C:\Program Files\Sun\JavaES5`.

5. Modify the following properties to reflect the host and port number of a consumer Directory Server installed in "Configuring For Replication" on page 69 .

   - `com.iplanet.am.directory.host`
   - `com.iplanet.am.directory.port`

6. Modify the following property to reflect the number of times Access Manager should continue to make the same request when the requested entry is not found.

   `com.iplanet.am.replica.retries`

7. Modify the following property to reflect the number of milliseconds Access Manager should allow to elapse between retries.

   `com.iplanet.am.replica.delay.between.retries`

8. In each Access Manager Authentication module enabled, use the Access Manager Console to specify the consumer directory installed in "Configuring For Replication" on page 69:

   - For the first LDAP server and port, specify the host name and port number for the primary (consumer) Directory Server. For example: `consumer1.example.com:389`.
   - For the second LDAP server and port, specify the host name and port number for the secondary (or supplier) Directory Server. For example, `supplier1.example.com:389`.

9. In the `serverconfig.xml` file, specify the host name and port number of the consumer directory installed in "Configuring For Replication" on page 69, as shown in the following example for the `serverconfig.xml` file.

10. Restart Access Manager by restarting the web container.

Example of the `serverconfig.xml` File

The following example shows the `serverconfig.xml` replication modification.

```
<iPlanetDataAccessLayer>
<ServerGroup name="default" minConnPool="1"
maxConnPool="10">
<Server name="Server1"
host="consumer1.example.com" port="389"
type="SIMPLE" />
```

Configuring With a Load Balancer

The following figure shows a multiple-supplier configuration that includes Directory Proxy Server or a hardware load balancer. This configuration takes advantage of Access Manager support for failover, high availability, and managed load-balancing.

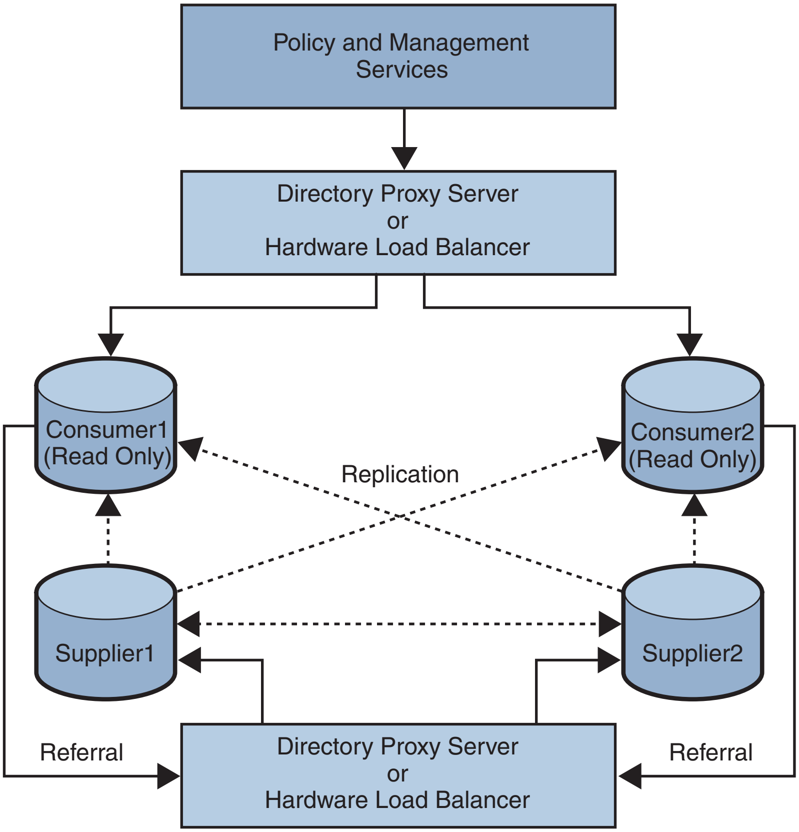

Policy and Management Services
Directory Proxy Server or Hardware Load Balancer
Consumer1 (Read Only)
Consumer2 (Read Only)
Replication
Supplier1
Supplier2
Referral
Directory Proxy Server or Hardware Load Balancer
Referral

FIGURE 5–4 Multiple-Supplier Configuration With a Load Balancer

Using LDAP load balancers adds a layer of high availability and directory failover protection beyond the level that is available with Access Manager. For example, Directory Proxy Server can specify the percentage of the load that gets redistributed to each server. And, if all back-end LDAP servers become unavailable, Directory Proxy Server continues to manage requests, rejecting client queries. If you install a load balancer, Access Manager must be configured to recognize the application.

1. Before configuring Access Manager, Set up the Directory Servers for replication. For information about directory replication and for detailed setup instructions, see the Sun Java System Directory Server documentation: `http://docs.sun.com/coll/1224.1`.
2. Install and configure the LDAP load balancer. Follow the instructions in the documentation that comes with the load balancer you are using.
3. In the `AMConfig.properties` file, modify the `com.iplanet.am.directory.host` and `com.iplanet.am.directory.port` properties to point to the load balancer host and port number of a consumer Directory Server.
4. For each Access Manager Authentication module enabled, use the Access Manager Console to specify the consumer Directory Server. In the following steps, the LDAP Authentication module is used as an example:
   - For the first LDAP server and port, type the host name and port number for the primary (consumer) Directory Server using the form `proxyhostname:port`.
   - Do not enter anything for the second LDAP Server and Port.
5. In the `serverconfig.xml` file, specify the host name and port number of the consumer Directory Server, as shown in the following example for the `serverconfig.xml` file.
6. Restart Access Manager by restarting the web container.

Load Balancer Modification to the `serverconfig.xml` File

The following example shows the load balancer modification to the `serverconfig.xml` file.

```
<iPlanetDataAccessLayer>
<ServerGroup name="default" minConnPool="1"
maxConnPool="10">
<Server name="Server1"
host="idar.example.com" port="389"
type="SIMPLE"
```

Directory Server With a Firewall

If your deployment is configured with a firewall between Access Manager and Directory Server, Access Manager connections can time out if the firewall idle connection timeout value is less than the Directory Server idle connection timeout value (`nsslapd-idletimeout` attribute). This problem usually occurs during non-peak usage hours when the load on Access Manager is low.

When Directory Server connections are dropped by the firewall, Access Manager does not recognize that the connections have been dropped and then goes through the pool of LDAP connections until all connections are exhausted. Access Manager must be restarted to create a fresh pool of LDAP connections. To prevent this problem, consider the following solutions:

- "Setting the Global Timeout Attribute" on page 74
- "Setting the Timeout Value for Individual Client Connections" on page 74

Setting the Global Timeout Attribute

You might be able to set the Directory Server global `nsslapd-idletimeout` attribute to a value less than the firewall idle connection timeout value. However, this solution might not be acceptable because `nsslapd-idletimeout` is a global configuration attribute that affects applications other than Access Manager.

Setting the Timeout Value for Individual Client Connections

Directory Server allows you to set specific attributes for individual client connections. The `nsIdleTimeout` attribute specifies the idle connection timeout value for individual clients. This value takes precedence over the `nsslapd-idletimeout` value set for the global Directory Server configuration.

Set the `nsIdleTimeout` attribute for the Access Manager user that binds to the LDAP directory, which by default is `amldapuser`. This attribute also applies to the `dsameuser` and `puser` users.

To add the `nsIdleTimeout` attribute for `amldapuser`, use either the Directory Server Console or the `ldapmodify` tool. For example:

```
ldapmodify -h host-name -p port
-D "cn=Directory Manager" -w password
dn: cn=amldapuser,ou=DSAME Users, dc=example,dc=com
changetype: modify
add: nsIdleTimeout
nsIdleTimeout: timeout-value
```

For *timeout-value*, specify a value less than the connection idle timeout value set for the firewall. Thus Directory Server will close the Access Manager connections for `amldapuser` before they are closed by the firewall.

To add the timeout for `dsameuser` or `puser`, use the above syntax, except set the `dn` option to the `dsameuser` or `puser` user.

The `com.sun.am.event.connection.idle.timeout` property in the `AMConfig.properties` file specifies the timeout value in minutes after which persistent searches will be restarted. This property ensures that persistent searches are restarted when the connections are dropped. Ideally, this value should be lower than the load balancer or firewall TCP timeout value, to make sure that persistent searches are restarted before the connections are dropped. A default value of zero (0) specifies that these searches will not be restarted.

For information about the Directory Server attributes and the `ldapmodify` tool, see the Sun Java System Directory Server documentation: `http://docs.sun.com/coll/1224.1`.

◆ ◆ ◆ CHAPTER 6

Implementation of an Access Manager Design

During the implementation phase of the solution life cycle you work from specifications and plans created during the deployment design phase to build and test the deployment, ultimately rolling out the deployment into production.

Note – Implementation information about from the previous version of the *Access Manager 7 2005Q4 Deployment Planning Guide* is now available in the *Sun Java System Access Manager 7.1 Postinstallation Guide*.

Implementation Phase Tasks

The implementation phase of the solution life cycle includes the following tasks:

- Determining and building the network and hardware infrastructure
- Installing and configuring Access Manager and related software according to an installation plan
- Migrating Access Manager data from existing applications to the current deployment
- Implementing an Access Manager user management plan
- Designing and deploying Access Manager pilots or prototypes in a test environment according to a test plan
- Designing and running functional tests and stress tests according to a test plan
- Rolling out the Access Manager test environment to a production environment according to a rollout plan
- Training Access Manager administrators and users of the deployment according to a training plan

APPENDIX A

Installed Product Layout

This appendix describes the directory layout after you install Sun Java™ System Access Manager 7.1 using the Sun Java System Enterprise System (Java ES) installer.

If you are deploying an Access Manager 7.1 WAR file, see Chapter 12, “Deploying Access Manager as a Single WAR File,” in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

The following table shows a summary of the Access Manager default directories after installation.

Summary of Access Manager Directories

TABLE A–1 Summary of Access Manager Directories

Description	Default Directory
Base Installation Directory See “Base Installation Directory” on page 80.	Solaris systems: `/opt/SUNWam` Linux and HP-UX systems: `/opt/sun/identity` Windows systems: `C:\Program Files\Sun\JavaES5\identity` During installation, you can specify a different base installation directory for `/opt`, `/opt/sun`, or `C:\Program Files\Sun\JavaES5`, if you prefer. However, do not change the `/SUNWam`, `/identity`, or `\identity` product directory name.
Configuration Directory See “Configuration (`/config`) Directory” on page 86.	Solaris systems: `/etc/opt/SUNWam/config` Linux and HP-UX systems: `/etc/opt/sun/identity/config` Windows systems: `C:\Program Files\Sun\JavaES5\identity\config`

TABLE A–1 Summary of Access Manager Directories *(Continued)*

Description	Default Directory
Temporary Files Directory	Solaris systems: `/var/opt/SUNWam/tmp` Linux and HP-UX systems: `/var/opt/sun/identity/tmp` Windows systems: `C:\Program Files\Sun\JavaES5\identity\tmp`
Debug Files Directory	Solaris systems: `/var/opt/SUNWam/debug` Linux and HP-UX systems: `/var/opt/sun/identity/debug` Windows systems: `C:\Program Files\Sun\JavaES5\identity\debug`
Log Files Directory	Solaris systems: `/var/opt/SUNWam/logs` Linux and HP-UX systems: `/var/opt/sun/identity/logs` Windows systems: `C:\Program Files\Sun\JavaES5\identity\logs`

Base Installation Directory

The default base installation directory depends on the platform where you are installing Access Manager:

- Solaris systems: `/opt`
- Linux and HP-UX systems: `/opt/sun`
- Windows systems: `C:\Program Files\Sun\JavaES5`

In the Access Manager documentation, the *AccessManager-base* variable represents the base installation directory for Solaris, Linux, and HP-UX systems. For Windows systems, the *javaes-install-dir* variable represents the Java ES 5 installation directory.

Within the base installation directory, Access Manager packages, shared binary files, command-line tools, and other files are installed in the `/SUNWam` directory on Solaris systems, the `/identity` directory on Linux and HP-UX systems and `\identity` directory on Windows systems. Therefore, the default base and product directory also depend on the platform:

- Solaris systems: `/opt/SUNWam`
- Linux and HP-UX systems: `/opt/sun/identity`
- Windows systems: `C:\Program Files\Sun\JavaES5\identity`

Note – During installation, you can specify a different base installation directory if you wish. However, do not change the `/SUNWam`, `/identity`, or `\identity` product directory name.

On Windows systems, the \setup directory contains the following files that you can use to configure Access Manager:

- amconfig.bat is a batch file used to deploy, configure, and reconfigure Access Manager. This file is equivalent to the amconfig script on UNIX and Linux platforms.
- AMConfigurator.properties is the configuration input file that contains the Access Manager configuration properties. This file is equivalent to the amsamplesilent file on UNIX and Linux platforms. The values in AMConfigurator.properties should not contain backslashes (\).

The /SUNWam, /identity, or \identity directory contains the following files and directories:

- Web application archive (WAR) files, such as amcommon.war, amconsole.war, ampassword.war , and amserver.war.

  For information about WAR files, see the *Sun Java System Access Manager 7.1 Developer's Guide* and Chapter 12, "Deploying Access Manager as a Single WAR File," in *Sun Java System Access Manager 7.1 Postinstallation Guide*.

  Subdirectories include:

  - "/bin Directory" on page 81
  - "/docs Directory" on page 83
  - "/dtd Directory" on page 83
  - "/include Directory" on page 84
  - "/ldaplib Directory" on page 84
  - "/lib Directory" on page 84
  - "/locale Directory" on page 84
  - "/migration Directory" on page 84
  - "/public_html Directory" on page 84
  - "/samples Directory" on page 85
  - "/share Directory" on page 85
  - "/upgrade Directory" on page 85
  - "/web-src Directory" on page 85

After installing Access Manager, check the package installation accuracy by using the pkgchk (1M) utility. For example:

```
pkgchk -l -p /opt/SUNWam
```

/bin Directory

The following table describes the command-line tools and utilities in the /bin directory. For information about running these tools and utilities, see the *Sun Java System Access Manager 7.1 Administration Reference*.

TABLE A–2 Access Manager Command-Line Tools and Utilities

Utility	Description
am2bak am2bak.bat (Windows)	Backs up the Access Manager components.
amadmin amadmin.bat (Windows)	Load XML service files into Directory Server and performs batch administrative tasks on the DIT.
amsfo, amsfoconfig, amsfopassword amsfo.pl, amsfoconfig.bat, amsfopassword.bat (Windows)	Access Manager session failover scripts.
ampassword ampassword.bat (Windows)	Changes passwords for Access Manager administrator or users.
amsamplesilent	Sample silent install file for use with the installation and configuration scripts.
amconfig, amutils, amdsconfig, amsdkconfig, amsvcconfig, amas70config, amwas51config, amwl81config, amws61config	Installation and configuration scripts for installing, configuring, and uninstalling Access Manager instances. For information about these scripts, see Chapter 2, “Running the Access Manager amconfig Script,” in *Sun Java System Access Manager 7.1 Postinstallation Guide*.
amserver	Start and stops the amunixd and amsecuridd daemons.
amtune directory	Contains the Access Manager tuning scripts, which allow you to set operating system, Access Manager, web container, and Directory Server parameters to improve performance.
amverifyarchive amverifyarchive.bat (Windows)	Verifies the log archives to detect possible tampering and/or deletion of any files in the archive.
bak2am bak2am.bat (Windows)	Restores Access Manager components backed up by the am2back or am2back.bat utility.
ldapmodify	Edits the contents of an LDAP directory, either by adding new entries or by modifying existing ones.
ldapsearch	Issues search requests to an LDAP directory and displays the result as LDIF text.
amGenerateLDIF.pl and amGenerateNI.pl	Access Manager bulk federation scripts.

TABLE A–2 Access Manager Command-Line Tools and Utilities *(Continued)*

Utility	Description
`am2bak.template`, `amserver.template`, `amadmin.template`, `amverifyarchive.template`, `ampassword.template`, and `bak2am.template`	Access Manager template files.

`/docs` Directory

The `/docs` directory contains the HTML, JAR, CSS, and related files used for the Java API reference (Javadocs).

`/dtd` Directory

The `/dtd` directory contains the Document Type Definition (DTD) files used by Access Manager. A DTD defines the structure for XML files accessed by Access Manager. For more information, see the *Sun Java System Access Manager 7.1 Developer's Guide*.

The following table describes the Access Manager DTD files in the `/dtd` directory.

TABLE A–3 Access Manager DTD Files

File	Description
`Auth_Module_Properties.dtd`	Defines the structure for XML files used by the authentication modules to specify their properties.
`amAdmin.dtd`	Defines the structure for XML files used to perform batch LDAP operations on the directory tree using the `amAdmin` command-line tool.
`amWebAgent.dtd`	Defines the structure for XML files used to handle requests from, and send responses to, web agents. This file is deprecated and remains for purposes of backward compatibility.
`policy.dtd`	Defines the structure for XML files used to store policies in Directory Server.
`remote-auth.dtd`	Defines the structure for XML files used by the Authentication Service's remote Authentication API.
`server-config.dtd`	Defines the structure for `serverconfig.xml` which details ID, host and port information for all server and user types.
`sms.dtd`	Defines the structure for XML service files.

TABLE A–3 Access Manager DTD Files *(Continued)*

File	Description
`web-app_2_2.dtd`	Defines the structure for XML files used by the Access Manager deployment container to deploy J2EE applications.

`/include` Directory

The `/include` directory contains header (`.h`) files.

`/ldaplib` Directory

The `/ldaplib/ldapsdk` subdirectory contains the shared object (`.so`) files needed to run the LDAP utilities included with Access Manager.

`/lib` Directory

The `/lib` directory contains JAR files and additional shared object (`.so`) files. It also contains a link to the `AMConfig.properties` file.

`/locale` Directory

The `/locale` directory contains the localization properties files. Each properties file includes a corresponding English localization file. For example, `amAdminCLI_en.properties` is the corresponding file for `amAdminCLI.properties`.

`/migration` Directory

The `/migration` directory contains the scripts and supporting files used to migrate data from earlier versions of Access Manager.

For more information about migration, the *Sun Java Enterprise System 5 Upgrade Guide* in the following documentation collection: `http://docs.sun.com/coll/1286.2`.

`/public_html` Directory

The `/public_html` directory and subdirectories contain the HTML and related files used for the Access manager Console online help.

/samples Directory

The /samples directory contains the following subdirectories: /admin, /appserver, /authentication, /console, /csdk, /liberty, /logging, /phase2, /policy, /saml, /sso, and /um.

Each subdirectory contains samples for the respective functionality, which is indicated by the subdirectory name. For more specific information about these samples, see the Readme.html file.

/share Directory

The /share/bin subdirectory contains the following additional utilities used internally by Access Manager, including amsecuridd, amunixd, amwar, checkport, and wsutils.ksh.

/upgrade Directory

The /upgrade directory contains the following directories:

- The /scripts contains the upgrade scripts and files.
- The /services directory contains directories for the Access Manager services.

/web-src Directory

The /web-src directory contains the subdirectories in which Access Manager J2EE web applications are deployed on a web container. It contains the following subdirectories:

- applications/ directory where the Access Manager Console is deployed. It contains the index.html file and various subdirectories. The /console directory contains various console related subdirectories.
- The /common directory (and subdirectories) is where the Access Manager Liberty Common Domain component is deployed.
- The /password directory (and subdirectories) is where the Access Manager Password Synchronization component is deployed. It contains the index.html file and the various subdirectories.
- The /services directory (and subdirectories) is where Access Manager Core Services are deployed. It contains the index.html file and the various subdirectories.

Configuration (/config) Directory

The default location of the configuration (/config) directory depends on the platform where Access Manager is installed:

- Solaris systems: /etc/opt/SUNWam/config
- Linux and HP-UX systems: /etc/opt/sun/identity/config
- Windows systems: C:\Program Files\Sun\JavaES5\identity\config

The /config directory contains configuration, XML, and LDIF files, including:

- The .version file contains the current version of Access Manager.
- The AMConfig.properties and SSOConfig.properties files contain Access Manager configuration attributes.
- The serverconfig.xml file provides configuration information for the Access Manager for Directory Server.
- The /ldif subdirectory contains the LDIF files needed for populating the Directory Server data store when installing Access Manager. For example:
  - During installation, the ds_remote_schema.ldif file loads the Access Manager specific LDAP schema object classes and attributes (such as the iplanet-am-managed-people-container) needed to store Access Manager data in Directory Server. The sunone_schema2.ldif file loads the Access Manager specific LDAP schema object classes and attributes.
  - During uninstallation, The ds_remote_schema_uninstall.ldif file removes the Access Manager LDAP schema object classes and attributes from Directory Server.
- The /xml subdirectory contains XML files.
- The /ums subdirectory contains XML files, including:
  - The amserveradmin script loads the Access Manager services.
  - The ums.xml file provides a set of templates that contain LDAP configuration information for objects managed using Access Manager.
  - The XML files are generally not used for configuration. If they are modified, they must be manually reloaded into the Directory Server data store. (Any changes in the server are not synchronized with these files.) For information about the XML files in this directory, see the *Sun Java System Access Manager 7.1 Developer's Guide*.

Index