RSA® Access Manager 6.2 SP1 Servers Administrator's Guide

Contact Information

Go to the RSA corporate website for regional Customer Support telephone and fax numbers:
www.emc.com/domains/rsa/index.htm

Trademarks

RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of EMC trademarks, go to **www.emc.com/legal/emc-corporation-trademarks.htm#rsa**.

License Agreement

This software and the associated documentation are proprietary and confidential to EMC, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability. This software is subject to change without notice and should not be construed as a commitment by EMC.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

December 2013

Contents

Preface

About this Guide

This guide describes how to administer RSA Access Manager 6.2 SP1 using the Administrative Console. This guide is intended for security administrators and other trusted personnel. Do not make this guide available to the general user population.

RSA Access Manager Documentation

For more information about RSA Access Manager, see the following documentation:

Release Notes. Provides information about what is new and changed in this release, as well as workarounds for known issues. The latest version of the *Release Notes* is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Getting Started. Lists what the kit includes (DVD, CDs, licenses and documentation), specifies the location of documentation on the DVD, and lists RSA Customer Support web sites.

Planning Guide. Provides a general understanding of RSA Access Manager, its high-level architecture, its features, and deployment information.

Servers Installation and Configuration Guide. Provides instructions for installing and configuring the RSA Access Manager Servers and additional components. This guide also contains descriptions for different configuration options, features, and production environment considerations.

Administrator's Guide. Provides information for security administrators about using the RSA Administrative Console to administer users, resources, and security policy in RSA Access Manager.

Developer's Guide. Provides information about developing custom programs using application programming interfaces (APIs) included with the RSA Access Manager Servers.

API Delta Document. Provides information about the differences between previous and current versions of the APIs included with the RSA Access Manager Servers.

Upgrade Guide. Provides information about how to upgrade from previous versions of the RSA Access Manager Servers, data store schema, and data to the current version.

RSA Administrative Console Help. Provides instructions on how to perform specific administrative tasks. To view Help, click the **Help** tab on the RSA Administrative Console.

RSA Access Manager User Self-Service Console Help. Describes day-to-day user tasks performed in the User Self-Service Console. To view Help, click the **Help** tab on the RSA User Self-Service Console.

Related Documentation

For more information about products related to RSA Access Manager 6.2 SP1, see the following:

RSA Access Manager Agents documentation set. The documentation related to agents is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

RSA Adaptive Authentication documentation set. The documentation related to RSA Adaptive Authentication is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

RSA Envision documentation set. The documentation related to RSA Envision is available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Getting Support and Service

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
Customer Support Information	**www.rsa.com/support**
RSA Secured Partner Solutions Directory	**www.rsasecured.com**

RSA SecurCare Online offers a knowledgebase that contains answers to common questions and solutions to known problems. It also offers information on new releases, important technical news, and software downloads.

The RSA Secured Partner Solutions Directory provides information about third-party hardware and software products that have been certified to work with RSA products. The directory includes Implementation Guides with step-by-step instructions and other information about interoperation of RSA products with these third-party products.

Before You Call Customer Support

Make sure you have direct access to the computer running the RSA Access Manager software.

Please have the following information available when you call:

- ❑ Your RSA Customer/License ID.

  This is a paper license. You can find this number only on the license distribution medium. If you do not have access to the paper-based RSA Customer/License ID, contact RSA Customer Support.
- ❑ RSA Access Manager software version number and patch level.
- ❑ The make and model of the machine on which the problem occurs.
- ❑ The name, version, and patch level of the operating system under which the problem occurs.

1 Preparing for Administration

This chapter introduces you to RSA Access Manager 6.2 SP1 administration, and includes information that you must know before you administer Access Manager. It includes an overview of Access Manager terms and concepts, an administrator's task list, and an overview of the Access Manager Administrative Console.

Before You Begin

Before you administer RSA Access Manager:

- Become familiar with Access Manager concepts and terminology. For more information, see the following section, Administrative Terms and Concepts. Once you familiarize yourself with these administration concepts, see the RSA Administrative Console Help for instructions on performing specific tasks.
- Become familiar with the tasks that you can perform using the Administrative Console. For more information, see "Administrator Task List" on page 9.
- Know the default authorization mode. The default authorization mode is set during the Authorization Servers configuration process. The default authorization mode controls access to resources not explicitly protected in the Administrative Console. For more information, see "Authorization Mode" on page 41.

Administrative Terms and Concepts

This section describes important Access Manager administrative terms and concepts.

Delegated Administration

Delegated administration allows administrative tasks to be shared among several individuals:

Administrative Groups. An administrative group is a collection of objects, such as users, user groups, servers, applications, and properties. Administrative groups are managed by administrators who are assigned a role in the administrative group.

Roles. Roles define the actions administrators can perform on the objects owned by their administrative groups. Each administrator must be assigned at least one role. For more information about roles, see "Roles" on page 20.

Administrators. Access Manager administrators manage objects and security policies for their administrative groups. Each administrator has one or more roles, each of which lets the administrator manage a set of Access Manager objects. You can have as many administrators as your organization requires. For more information, see "Password Policies" on page 22.

Secure Delegated Impersonation

Secure delegated impersonation lets you delegate administrators to act as impersonators and access the applications used by other users and troubleshoot the issues that the users face. When you enable secure delegated impersonation, details of the impersonated user and resources accessed by the impersonator are logged in the Authorization Server log file.

Secure delegated impersonation policies can be set for applications, application functions, and resource URL, using Access Manager Administrative APIs. For more information, see the *Developer's Guide*.

Users and User Groups

Access Manager uses an external data store such as LDAP or SQL to store user information. Before you can grant access to a user, information about the user must be added to the data store.

To make granting and denying of access to resources easier, and to help you organize your Access Manager system, you can create user groups. A user group can contain users called member users, and other user groups called member groups.

For more information about managing users and user groups, see Chapter 3, Administering Users and User Groups.

In addition to the required information you store in your user accounts, you can assign properties to users. Properties are customizable fields that let you store organization specific data with your user records. Properties also serve as evaluation criteria for Smart Rules. For more information, see "Properties" on page 33.

Resources

Objects that you protect with Access Manager are called resources. A resource can be an individual file, such as an image file, an entire directory on your web or application server, or an entire application.

In Access Manager, an application is a grouping of one or more associated resources. When you want to add a resource to Access Manager, you add it to an application.

For more information about resources, see Chapter 4, Adding and Managing Resources.

Security Policies

Security policies are the link between users and resources. There are two types of security policies in Access Manager:

Smart Rules. Dynamic access control policies that allow or deny a user access to resources based on the value of a user's properties.

Entitlements. Permissions that allow or deny access to a specific resource or application for a specific user or user group.

For more information, see Chapter 5, Configuring Security Policies.

Administrator Task List

The following table lists the basic tasks required of Access Manager and also provides a brief description of the issues to be considered before performing these tasks.

Task	Considerations
Delegated Administration	
Add administrative groups. See "Administrative Groups" on page 17.	Decide how to organize your groups. For example, geographical divisions or business units.
Add roles. See "Roles" on page 20.	Decide the scope of each administrator's privileges.
Create administrators. See "Administrators" on page 15.	Decide how many administrators you need to manage your administrative groups.
Create password policies. See "Password Policies" on page 22.	Decide what rules you want to enforce related to user passwords.
Users and User Groups	
Add Properties. See "Properties" on page 33.	Determine your need to store additional user information, and decide if you want to use Smart Rules to control user access to resources.
Add users. See "Users" on page 29. **Note:** Ensure you are not accessing user and user group data in read-only mode.	You can set property values as you create users.

Task	Considerations
Add user groups. See "User Groups" on page 39. **Note:** Ensure you are not accessing user and user group data in read-only mode.	Your user group structure depends on your organization's structure, as well as your plans to use entitlements to allow or deny group access to resources.
Resources	
Add servers. See "Servers" on page 43.	You must have a valid hostname and port number. Web and application servers name must match values of the cleartrust.agent.web_server_name and the cleartrust.agent.server_name parameters respectively in the Server configuration file.
Add applications. See "Applications" on page 44.	The organization of applications, reflects your planning of logically related resources.
Add resources. See "Resources" on page 45.	Add an HTTP or J2EE resource to one of the applications you create.
Add functions. See "Functions" on page 51.	To protect resources that are not web-based, collaborate with developers to protect specific functions.
Security Policy	
Add Smart Rules. See "Smart Rules" on page 56.	Create Smart Rules based on the properties you define.
Test Smart Rules. See "Testing Security Policies" on page 63.	Test Smart Rules prior to implementation.
Add entitlements. See "Entitlements" on page 53.	Create entitlements to control access based on user identity or user group membership.
Test entitlements. See "Testing Security Policies" on page 63.	Test entitlements prior to implementation.

RSA Access Manager Administrative Console

All Access Manager administrative tasks are performed using the Administrative Console, an application that you access from a web browser. The Administrative Console is a web-based, Java Server Page (JSP) application that you install on any supported application server or servlet engine. For a list of supported application servers and servlet engines, see the *Servers Installation and Configuration Guide*. The Administrative Console includes Online Help, which provides instructions for performing the tasks described in this guide.

Note: Tasks that can be performed using the Administrative Console can also be done using the Access Manager Administrative APIs, which have larger capabilities than those provided by the Administrative Console. For more information, see the *Developer's Guide*.

Note: The Administrative Console is set to time out after 10 minutes. If you begin a lengthy process, the Administrative Console may time out and close before the process is finished. The process, however, continues and finishes on the servers.

Logging On to the RSA Administrative Console

Before logging on to the Administrative Console, make sure that you have:

- URL of the Administrative Console.
- Your User ID and password.

Note: The autocomplete feature is disabled by default in Access Manager.

To log on to the Administrative Console:

1. Go to the following URL:
   http://<*fully qualified domain name*>:<*port number*>/axm-admin-gui
2. Enter your User ID and password, and click **Log On**.

Accessing Pages in the RSA Administrative Console for the First Time

If your Administrative Console is installed on an Apache Tomcat or an IBM WebSphere Application Server, the Administrative Console JSP pages are not pre-compiled when the application is initially deployed. As a result, page downloads can be slow when accessed for the first time.

If you are waiting for a page to compile and download, and meanwhile attempt to go to another page that has not yet been compiled, the system may log you off from the Administrative Console.

Note: For Administrative Console installations on Oracle WebLogic Server, Administrative Console pages are pre-compiled.

Searching for Objects

The Administrative Console provides the ability to search for objects, such as administrative groups, users, or applications, in the Access Manager system.

Sorting Search Results

On pages where search results are displayed, the results can be sorted, to make searching for a particular object more efficient. Clicking an underlined column header title switches the display between ascending and descending order. By default, search results display in ascending order according to the contents of the first sortable column. Search results can be resorted by different columns, such as Last Name, First Name, or Administrative Group, by clicking the underlined column header title. For more information, see the Administrative Console Help topic "Sorting Search Results."

Case Sensitivity in Searches

Searches for most administrative objects (for example, users and groups) can be case-sensitive depending on your data store and its settings. The following table lists the default settings. To find out the actual setting of your data store, contact your data store administrator.

Data Store	Default Case Sensitivity
LDAP	No
Oracle	Yes
SQL Server	No

Note: In the *Administrator's Guide* and Help, all references to LDAP data stores, refer to all the directory servers, including Active Directory and Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Services (AD LDS).

Language Support

The Administrative Console has a menu for changing the language setting. To locate this menu, click **Options > Language** on any page of the Administrative Console. Access Manager currently supports United States English only. Future releases may support other languages.

Authorization Server Cache

In Access Manager, the Authorization Server caches user entitlement data, as well as resource and security policy data. To improve response time, runtime requests are verified, if applicable, against cached data instead of the data stores.

When you update records in the Administrative Console, those records are automatically refreshed in the cache.

Clear Cache Command

The Authorization Server cache is usually managed by configuration settings. There are occasions when you may need to manually clear the cache using the Administrative Console. Performing this command can have a profound effect on system performance. RSA recommends that you do not clear the cache unless you fully understand the consequences. For more information, see the Administrative Console Help topic "Clearing the Cache."

Secure Sockets Layer Connections

For added security, you can make Secure Sockets Layer (SSL) connections between the Administrative Console and the Entitlements Server.

Data Refreshing

The Administrative Console does not provide real time refreshing of data. Therefore, if information changes while you are working in the system, you do not automatically see the updated data. Rerun your queries to retrieve the latest data from your data store.

Similarly, the permissions of an administrator who is logged on to the Administrative Console are not updated until the next logon.

For example, suppose you removed the permission of Administrator "A" to add new users. If Administrator "A" was logged on to the Administrative Console at the time of the update, Administrator "A" can still add new users until the next logon.

2 Configuring Delegated Administration

RSA Access Manager uses delegated administration, which lets you designate multiple administrators with varied permissions and responsibilities, to manage your Access Manager system.

Administrators

You can assign administrative privileges to users and make them administrators. You can do this on the Add a New User or Edit User page.

Administrator privileges are determined by the roles assigned to them. For example, a role can let an administrator add and edit, but not delete users in a specific administrative group.

As an optional privilege, you can assign one of the privileges to any Access Manager administrator:

- Super Admin
- Help Desk Admin
- Config Admin
- Audit Admin

Note: All administrators must have at least one role assigned to them.

Super Admins

The Super Admin privilege gives an administrator the highest level of administrative privilege. Super Admins can view, add, edit, or delete any object in the Access Manager system, including other administrators, regardless of the administrative group that owns the object.

When you install the Administrative Console, an initial administrator account with Super Admin privileges is automatically created. You are asked to name this Super Admin account during installation of the Access Manager server. This Super Admin account is the starting point for setting up your Access Manager administration model.

Access Manager can function with only one administrator, the Super Admin. However, delegated administration can relieve the Super Admin of many administrative tasks associated with controlling access to system resources. Before you can assign the Super Admin privilege to another administrator, you must assign at least one other role to that administrator.

Important: It is possible that the administrators can be locked out of the system. This is because the administrators are stored as users. RSA recommends to have more than one administrator with Super Admin privileges in the system.

Help Desk Admins

The primary purpose of the Help Desk Admin is to reset passwords in Access Manager. The Help Desk Admin has the ability to view all user accounts and to view and edit passwords, regardless of the administrative group that owns the user. The Help Desk Admin can also allow access to a user who has been locked out.

Config Admin

The Config Admin privilege allows the administrator to modify all the parameters in the encrypted server configuration files.

Audit Admin

The Audit Admin privilege allows the administrator to modify only the log related parameters in the encrypted server configuration files.

Note: The Config Admin and the Audit Admin privileges can be used only when you encrypt the server configuration files. For more information on encrypting, decrypting, and modifying the server configuration files, see the *Servers Installation and Configuration guide*.

Administrators as Users

Administrators are stored in the system as users and are assigned to an administrative group. Administrators must have unique names, they cannot share names with other users or administrators. A search performed on users, includes users and administrators, but a search performed on administrators is restricted to administrators only. The administrators can:

- Be members of one or more user groups.
- Be assigned entitlements.
- Have properties and be assigned property values.

For instructions on creating administrators, see the Administrative Console Help topic "Creating Administrators."

Setting Up Delegated Administration

A Super Admin uses the Administrative Console to:

- Create administrative groups containing objects, such as users, user groups, a password policy, and resources.
- Define roles that control how an administrator can manage Access Manager objects.
- Designate some users as administrators to help with administrative responsibilities.
- Assign privileges to each administrator. The different privileges that can be assigned are:
  - Super Admin
  - Help Desk Admin
  - Config Admin
  - Audit Admin
- Create parent-child relationships among administrative groups.

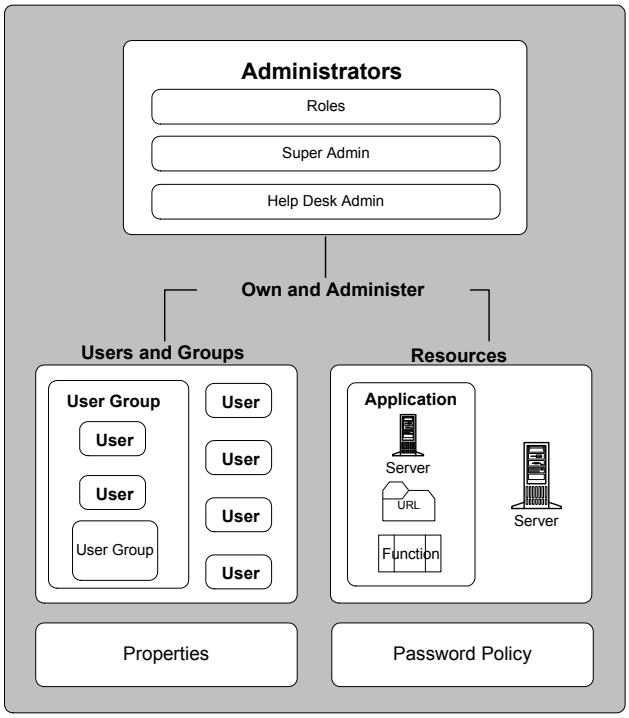
Administrative Groups

An administrative group is a collection of Access Manager objects, such as:

- Users
- User groups
- Applications
- Servers
- Properties
- Roles

The following figure depicts an administrative group.

Administrative Groups



These groups are created and managed using the Administrative Console. You can create several different administrative groups based on the needs of your organization. For example, administrative groups can be based on organizational structure and reflect departmental divisions, such as marketing, sales, shipping, and engineering. Administrative groups can also be based on geography with separate groups created for the regions or administrative centers of your organization.

An administrative group can own the users and user groups for a specific region, the web servers that house the region's resources, and any special web applications used within the region. Administrative groups can also include extranet partners, customers, and others external to the organization.

When you install Access Manager, the Default Administrative Group is created to contain and manage objects that are not being managed by another group.

Note: You cannot delete the Default Administrative Group.

Parent-Child Administrative Group Relationships

Administrative groups can be assigned parent-child relationships by the Super Admin, which allow greater ease of administration. For more information, see "Roles" on page 20.

Administrative Group Objects

When a Super Admin creates an administrative group, all objects assigned to the group are owned and managed in that group. For example, a user assigned to Group A can only be managed by an administrator with a role in Group A.

Transferring Object Ownership

To reorganize an administrative group, a Super Admin transfers one group's objects to another administrative group.

When deleting an administrative group, transfer all of the group's objects to another group before the group is deleted. If this is not done, ownership of the group's objects is automatically transferred to the Default Administrative Group.

For instructions, see the Administrative Console Help topic "Transferring Ownership by Administrative Group."

If you are using an LDAP data store, the default administrator created during Access Manager installation is not explicitly owned by any administrative group. Therefore, when you transfer ownership from the default administrative group to another administrative group, the default administrator is not transferred.

To transfer ownership of the default administrator, edit the **administrative group** field, and explicitly assign the default administrator to an administrative group. For instructions, see the Administrative Console Help topic "Editing Users."

Note: Transferring objects from a large administrative group can consume considerable system resources. RSA recommends that you perform this transfer during maintenance periods or off-peak hours.

Public and Private Objects

The visibility setting of an object controls whether an object, such as a user, application, or property is public or private, and lets you control administrators' access to sensitive information.

Public. Public objects can be viewed in the Administrative Console by administrators with a role in any administrative group. However, public objects can only be modified by Super Admins and administrators with an appropriate role in the group that owns the object.

Private. Private objects can be viewed in the Administrative Console only by Super Admins and by administrators who are assigned a role in the administrative group that owns the object.

Note: A Super Admin can grant a user access to private resources that do not belong to that user's administrative group. However, those resources and their associated access permissions are not visible to administrators of the user's administrative group.

You select a public or private visibility setting when you create the following objects:

- User
- User Group

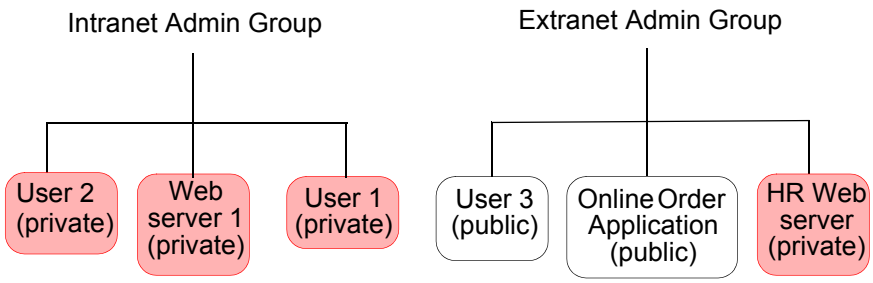
- Administrative Group
- Application
- Server
- Property

An object's visibility setting can be modified at any time.

When you create a new administrative group, you select whether you want the default visibility setting for objects assigned to that group to be public or private.

Example

In the following diagram, administrators in the Extranet Admin Group cannot see any of the objects owned by the Intranet Admin Group (User 2, Web server 1, and User 1) because these objects have been made private. At the same time, administrators in the Intranet Admin Group can see User 3 and Online Order Application owned by the Extranet Admin Group because these objects are public.



Roles

A role is a collection of privileges that control how an administrator manages Access Manager objects, such as users or applications in a particular administrative group. For example, a role might include the privilege to add new applications, but not the privilege to modify or delete them. Another role might include all of the privileges of a Super Admin except for the privilege to delete properties. Each administrative group can have several roles. Administrators can be assigned roles in more than one administrative group, allowing them to exercise privileges in several different groups. When administrators are assigned more than one role, they must choose one of their roles for the session, when they log on to the Administrative Console.

When you install Access Manager, the Default Administrative Role is created. By default, this role has no permissions. You can add permissions as necessary.

Each role is created for a single administrative group and can only be assumed within that group.

Roles are assigned when you create or edit administrators. All administrators must be assigned at least one role.

You can add the following privileges to a role:

- Add, edit, or delete roles
- Add, edit, or delete administrators
- Add, edit, or delete users
- Add, edit, or delete user groups
- Add, edit, or delete applications and resources
- Add, edit, or delete servers
- Edit user passwords
- Add, edit, or delete properties

RSA recommends that role names reflect an administrator's function in the organization, such as Help Desk, IT, or Human Resources. It is important that you select a unique name for each role and enter an explanation in the **Description** field. Identically named roles can confuse administrators when they try to assign roles to other administrators.

Normally, a role can only be exercised in the administrative group for which it was created. Therefore, if you want one administrator to manage several administrative groups, you must assign the administrator a role created for each group.

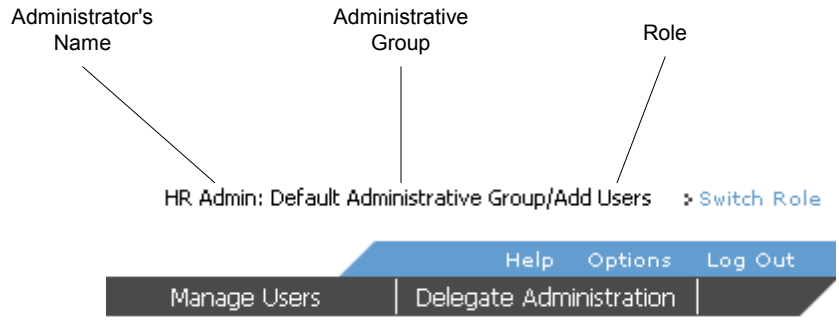
Normally, an administrator with roles in several groups can only administer one group at a time. To manage a different administrative group, an administrator must click **Switch Roles** on the Administrative Console, and select a role created for that group. The Switch Roles option is only available to administrators who have more than one role assigned to them.

Assigning a Role to More Than One Administrative Group

Though a role typically applies only to a single administrative group, it is possible for an administrator to assign the same role in more than one group. A role created for one group can be assigned in another group if the groups have a parent-child relationship. For example, if Group A is the designated parent of Group B, administrators in Group A who have logged on with a Group A role can also assign that role in Group B. This makes it possible for an administrator to manage objects in both Group A and Group B without having to switch roles. For instructions on creating a parent-child relationship between two groups, see the Administrative Console Help topic "Managing Child Administrative Groups."

Note: When you grant permission to add, edit, or delete administrators, you implicitly grant permission to add, edit, or delete users. This is because the administrators are Access Manager users. However, granting permission to add, edit, or delete users does not implicitly grant permission to add, edit, or delete administrators.

After you log on, your current role and administrative group are displayed on each page of the Administrative Console as shown below.



Password Policies

A password policy is a set of rules that establishes the required length of passwords, restricted characters and words, maximum password lifetime, password expiration dates, and user lockout rules. Once you create a password policy, you can assign it to any administrative group.

Each administrative group must be assigned a password policy. The password policy applies to all users within an administrative group. The Default Password Policy is created when you install Access Manager. If no password policy is assigned to an administrative group, this policy is applied. You can edit the Default Password Policy or create a new password policy to designate it as the default password policy. There must be a Default Password Policy for the system at all times. The default policy can be edited but cannot be deleted until you designate a new password policy as default.

You can create as many password policies as you need. When you design a password policy, balance the needs of your users with your security requirements. An excessively strict password policy, for example, one that requires overly long passwords or very frequent password changes may cause users to compromise security, mostly by writing down their passwords.

For instructions on adding a new password policy, see the Administrative Console Help topic "Adding Password Policies."

Important: If you use Active Directory, by default the **Password Lifetime**, **Password Lockout**, **Password History**, **Account Starts**, and **Account Expires** fields are disabled. Though you can configure Access Manager to support these features, RSA recommends you to use Windows mechanisms. For information about enabling Windows password expiration, see your Windows documentation.

If you are using an LDAP data store other than Active Directory and want to support the Password Lifetime, Password Lockout, Password History, Account Starts, and Account Expires features, your LDAP data store must apply the ctscUserAuxClass class to Access Manager users, and allow write access to user records.

Password Parameters

The different password parameters are:

- Lifetime
- History
- Minimum Lifetime
- Minimum Length
- Maximum Length
- Upper Case Character Count
- Lower Case Character count
- Special Character Count
- Numeric Character Count
- Excluded Characters
- Excluded Words File
- Non-Alpha Required
- Lock Out
- E-mail Notification

Lifetime (required)

Longer a password exists, it is more likely to be compromised. Access Manager lets you set a password lifetime rule, also called as password expiration rule, that forces the user to change password after a specified period of time has passed. When Access Manager expires a user's password, the user is locked out of all resources protected by Access Manager until a new password is selected.

The maximum value for the password lifetime setting is 106,751,991,167 days. By entering a large number in the **Lifetime** field (for example, 800,000 days) on the Edit Password Policy page, it is possible to set a password lifetime that is effectively permanent.

The following authentication systems do not let Access Manager expire passwords:

- LDAP authentication when users are stored in SQL.

  If your users are stored in an SQL user data store but authentication is done against an LDAP directory, Access Manager cannot expire user passwords. You must use your standard LDAP tools to reset passwords in LDAP.

- LDAP authentication when users are stored in Active Directory.

  Although you can configure Access Manager to support password expiration on Active Directory, RSA recommends that you use the Windows password expiration mechanism. For information about activating Windows password expiration, see your Windows documentation.

There are no password expiration limitations, if your Access Manager installation uses a Sun ONE Directory Server to store and authenticate users.

History (optional)

The password history is the number of recently used passwords that a user cannot reuse as the current password. For example, the three most recent passwords used by the user cannot be used again if the **Password History** field is set to three.

Minimum Lifetime (optional)

The **Minimum Lifetime** field lets you set the amount of time required between password changes. This field prevents a user from bypassing the **Password History** field by repeatedly changing the password. This feature is disabled by default with a setting of 0 seconds. To enable minimum lifetime, you need to enter a time value in this field that is shorter than the password lifetime. The maximum value for this setting is one year.

Minimum Length (required)

Passwords that are too short are vulnerable to attacks. Administrators must specify a minimum required length for user passwords. The default value is 8 characters.

Maximum Length (required)

Passwords that are too long can be difficult to remember. Administrators can specify a maximum length for user passwords. The default value is 32 characters.

Upper Case Character Count (optional)

Password having variety in characters are less vulnerable to attacks. This field specifies the least number of Upper case alphabetic characters required in the password. The default value is 0.

Lower Case Character Count (optional)

Password having variety in characters are less vulnerable to attacks. This field specifies the least number of Lower case alphabetic characters required in the password. The default value is 0.

Special Character Count (optional)

You can configure Access Manager Password policies to set the least number of special characters to be specified in this field. This policy enforces user to have special characters in the password equal to the number specified in this field. The default value is 0.

Numeric Character Count (optional)

Using Numeric Character Count field you can specify the least number of numeric characters required in the password. The default value is 0.

Excluded Characters (optional)

In Access Manager, you can enter a list of specific characters that cannot be used in passwords. For example, if you enter **# $ *** in the **Excluded Characters** field, the system rejects words such as "pass#word," "pa$$word," and "p*ssw*rd," and forces users to select a different password.

Excluded Words File (required)

The excluded words file contains words that cannot be used as passwords, including several thousand commonly used words, like "password," that are likely to be included as part of any dictionary attacks on the system. When users log on to Access Manager, their password is checked by the Entitlements Server against the excluded words file. You can customize the excluded words file located in the ***AXM_HOME*/conf** directory, where AXM_HOME is the location, where you have installed the Access Manager servers.

Non-Alpha Required (optional)

You can configure Access Manager password policies to require at least one non-alphabetic character. The most common attacks on passwords are dictionary attacks. Adding non-alphabetic characters to a password significantly enhances the security of a password, for example, changing "password" to "pa1ss2wo3rd".

Keep in mind that common alphanumeric substitutions, such as the numeral 1 for the letter L, the numeral 3 for the letter E and the numeral 7 for the letter T, have been integrated into many password cracking tools.

Lock Out (optional)

You can set Access Manager password policies to lock out user accounts when users make too many unsuccessful logon attempts. You can also set the system to automatically unlock each locked account after a specified period of time. If you do not enable the automatic unlock feature, only administrators with permission to edit user accounts can unlock a user account, and only administrators with permission to edit administrator accounts can unlock administrator accounts.

For password policies that lock out users after a specified number of invalid logon attempts, the counter that determines when a user is locked out is automatically reset after one of these events takes place:

- Reset time expires
- User's password is modified
- User is unlocked by an administrator
- User successfully authenticates within the reset time

In the last case, if the password policy locks out a user after three unsuccessful attempts, but the user authenticates after two unsuccessful attempts, the counter is reset.

For administrators, the counter is also reset when the administrator successfully logs on to the Administrative Console.

For more information, see the Administrative Console Help topic "Locking Out or Unlocking a User."

Note: If you use multiple authentication types in Access Manager, all logon failures of an user, regardless of authentication type, are counted in the same counter. For example, if a user fails using Basic authentication, and then fails using RSA SecurID authentication, the counter is set to two.

Access Manager does not interact with any logon failure counters that are resident in your underlying user authentication mechanisms.

E-mail Notification (optional)

When properly configured, Access Manager sends a notification e-mail to an administrator when a user is locked out of the system. If the e-mail server is unreachable, the e-mail is not sent and the Access Manager logs an error in its log file. This does not affect the Access Manager operation in any other way.

Lockout Mechanisms

Access Manager does not interact with the underlying lockout mechanism for the following authentication mechanisms:

- RSA SecurID authentication
- Sun ONE Directory Server authentication
- Certificate authentication

As a result, if a user is locked out using any of these authentication mechanisms, the user can still authenticate outside of Access Manager.

If you are using an Active Directory, Access Manager uses the underlying Active Directory mechanism to enforce user lockouts. As a result, if a user stored in Active Directory is locked out, the user is locked out completely and cannot access any resource either protected by Access Manager or that relies on Active Directory for authentication.

For users stored in other data stores, Access Manager uses its own, internal lockout mechanism. For example, if your user is stored in a Sun ONE Directory Server and the user is locked out of Access Manager, the user can still use the Sun ONE Directory Server user account for other purposes not related to Access Manager.

Manually Expiring a User's Password

A user password can be manually expired before the password lifetime expires. Manually expiring passwords forces users to change their passwords the next time they access the system. For example, when you create new user accounts, you can assign a default password, and this require users to immediately create a new password on their first logon.

Typically, users are presented with a "Password Expired" page after their passwords have expired.

You can expire passwords in the following ways:

- Set the user password expiration date to match the current date when you add or edit a user in the Administrative Console.

- Use the **Expire Now** button on the Add a New User or Edit User page. For instructions, see the Administrative Console Help topic "Forcing Password Expiration."

Note: Your changes apply to passwords for Basic authentication only. Windows NT and RSA SecurID authentication must be managed in the native Windows NT or RSA Authentication Manager environment.

Permission to Reset Passwords

You can reset user passwords only if you are an administrator who has been assigned one of the following:

- The Super Admin privilege
- The Help Desk Admin privilege
- A role that grants you permission to reset passwords, and is in the same administrative group as the user whose password you want to reset.
- A role that grants you permission to modify users, and is in the same administrative group as the user whose password you want to reset.

You can reset administrator passwords only if you are an administrator who has been assigned one of the following:

- The Super Admin privilege
- A role that grants you permission to reset passwords and is in the same administrative group as the administrator. You must also have more privileges than the administrator whose password you want to reset.

Note: Only a Super Admin can reset another Super Admin's password.

3 Administering Users and User Groups

RSA Access Manager controls access to your system resources based on user or user group identity, or on the values of properties assigned to selected users. This chapter describes Access Manager user and user group management concepts, including the creation and management of properties.

Users

Access Manager uses an external data store, such as Active Directory, ADAM, AD LDS, or SQL, to store user information. Each user account includes specific information about the user. You can also define properties, and other optional user information, such as a user's e-mail address.

You can manage users from the Manage Users menu of the Administrative Console or with the native data management tools of your data store.

Adding and Modifying User Information

You can use one of the following to add user information in the Access Manager system:

- The Administrative Console

  For more information, see the Administrative Console Help topic "Adding Users."
- The Native User Management Tool

  For more information, see your data store documentation.

  Note: If you are using an LDAP data store, RSA recommends that you use native administration tools to administer users and user groups.

RSA recommends that you use the same tool to edit data as you did to add the data to the system. For example, if you used the Administrative Console to add your data, use the Administrative Console to edit that data, and if you used your data store's native user management tool to add your user data, use that same tool to edit the data.

Note: Active Directory does not allow a user and a user group to have the same name. For example, if you have a user group named admin and attempt to create a user named admin, you will receive an error message.

User Attributes

This section describes the attributes, both required and optional, that describe an Access Manager user. If you store your users in an LDAP data store, see the Notes column in the following table for the default attribute names used in LDAP.

Field	Description	Notes
User ID	Logon ID for the user	Required. Can be from 1 to 255 characters. For Active Directory, the User ID can be no longer than 20 characters. In LDAP, the User ID cannot be modified. Special characters cannot be used. LDAP attribute name: cn. In Active Directory, you cannot have an User ID that is similar to an existing User Group name.
First Name	User's first name	Optional. Can be from 1 to 255 characters. LDAP attribute name: givenName.
Last Name	User's last name	Required. Can be from 1 to 255 characters. LDAP attribute name: sn.
E-mail	E-mail address for the user	Optional. Can be from 1 to 255 characters. LDAP attribute name: mail.
Certificate DN	User's distinguished name for certificate authentication	Optional. The DN of the client-side certificate for authentication must match this value. User searches with DN as a criterion may fail when certdn is configured to map to DN.
Properties	A property is a custom data field, such as department, that you define and in which you can store organization-specific user information. The primary purpose of properties is to create evaluation criteria for Smart Rules.	If you are using an LDAP directory as your data store, an attribute for the property must exist in your directory schema before you can add a property definition for it in Access Manager.

Field	Description	Notes
Account Starts	Date and time the user account becomes active	Default is the time on the machine where the application server is running when the Add a New User page is first called. Time zone is also determined by the machine where the application server is running.
Account Expires	Date and time the account expires	The default setting is one year after Account Starts.
Lock Out	Immediately disables any permissions granted to the users, and blocks them from accessing protected resources. Can also indicate if a user is locked out because of too many incorrect passwords entered.	Optional.
Administrator	Identifies the user as an administrator	Optional. Only if you want to create an admin user, select the **User is an RSA Access Manager administrator** checkbox. You must assign at least one role to an administrator.
Roles	An administrator's collection of privileges to create, edit, and delete Access Manager objects.	Optional. Only if you are creating an admin user you can assign roles to them.

Field	Description	Notes
Optional Privileges	You can select either of the following: Super Admin, which can perform any action on any object or resource. Help Desk Admin, which has the ability to change or reset passwords across all existing administrative groups. Config Admin, which can modify all the parameters in the encrypted server configuration files. Audit Admin, which can modify only the log related parameters in the encrypted server configuration files. **Note:** The Config Admin and the Audit Admin privileges can be used only when you encrypt the server configuration files. For more information on encrypting, decrypting, and modifying the server configuration files, see the *Servers Installation and Configuration guide*.	Optional. Administrators only. Only administrators with privileges to create or edit administrators can assign these privileges.
Status	Indicates whether the password is active or has expired.	Optional.
Password	The user's unique password for Basic authentication.	Required. This password is not used for Windows NT, RSA SecurID, or Certificate authentication. It is only used for Basic authentication. LDAP attribute name: userPassword.

Field	Description	Notes
Retype Password	Re-enter the password you chose for the user. Must be the same as the password you entered in the **Password** field.	Required.
Password Expires	The date on which the password expires.	The default password lifetime is determined by the password policy associated with the administrative group that owns the user. The time zone for the date is determined by local system time. Changing the date in the **Password Expires** field overrides the default password lifetime setting.
Administrative Group	Administrative group that owns the user account.	Required. The default is the administrative group of the administrator that created the user.
Visibility	Identifies the user as either public or private.	Required.

Duplicate Certificate DN Values

Access Manager does not check for duplicate certificate DN values when you create or modify users in an LDAP data store. If you want your LDAP directory to enforce uniqueness, use DN rather than ctscUserDN.

Properties

A property is a custom data field that you define, which is specific to your organization. Properties can include any data that your organization stores or maintains for its users, such as age, account status, department, date of hire, customer type, and so on. For example, if you want to add a data field to every user that indicates what region a user lives in, you can create a property called RegionCode.

Note: Properties must have unique names. They cannot have the same name as required user information fields.

You create properties in the Add a New Property page of the Administrative Console, and set property values for a given user on the Add a New User or Edit User page. You can also edit properties on the Edit Property page. To add a property to a large number of users, you can create the property in the Administrative Console, and use the Access Manager Administrative API to automatically enter values.

If you are using an LDAP data store, you must map each Access Manager property to an LDAP attribute with an identical name. If you are using an SQL data store, SQL automatically creates properties in the data store, as you create them in Access Manager. For information about mapping properties to LDAP attributes, see "Properties for Directory Servers: Mapping to Attributes" on page 37.

The primary purpose of a property is to serve as evaluation criteria for Smart Rules. For information on Smart Rules, see Chapter 5, Configuring Security Policies.

Note: If you plan to use a Smart Rule based on a property to control access to a resource when you set up your Access Manager system, RSA recommends that you define the property before you create any users. This lets you specify the value of that property for each user as you create user accounts.

You can also configure Access Manager to automatically publish selected properties to the HTTP header, which allows them to be read by any Access Manager Runtime API client program. For more information, see "Publishing Properties to the HTTP Header" on page 38.

For more information on properties, see the Administrative Console Help topic "Understanding Properties."

Multi-Value Properties

If you are using an LDAP data store, you can create properties with more than one value. For example, you can define a property called phone number and enter multiple phone numbers for each user.

When you create a multi-value property, you must map the property to a multi-value LDAP attribute.

Once you have created a property, you cannot edit the **Multi-Value** checkbox. To change a property to a multi-value property, delete the property and create a new multi-value property. Likewise, to change a multi-value property to a single-valued property, delete the multi-value property and create a new property.

External Properties

Access Manager also lets you define external properties. External properties are properties that have values stored in an external data store.

Once you create an external property, you cannot edit the **External** field. If you no longer want an external property to be external, delete the external property and create a new property. Also, because the value of an external property is stored in an external data store, you cannot view or edit the value of an external property with the Administrative Console.

Property Data Types

The following table describes available property data types. Select the relevant data type when you create a property.

Note: LDAP directories do not support the float data type.

Data Type	Description	Example	Format/Allowed Values
Boolean	True or False	Current depositor? External user?	True False
String	A character string	The user's street address	Any string
Integer	An integer	The user's level of security clearance	Minimum: -2147483648 Maximum: 2147483647
Float	A floating point decimal value	The user's account balance	Minimum: 1.40129846432481707e-45f Maximum: 3.40282346638528860e+38f
Date	A date	The user's birthday The user's retirement date	mmm-dd-yyyy

Property Fields

Properties are defined in the Administrative Console by the fields described in the following table. Required information is noted.

Field	Description	Notes
Property Name	Name for the property.	Required. Can be from 1 to 255 characters. If you are using a directory server as your data store, Property Name must be identical to a corresponding attribute in the directory schema. For more information, see "Properties for Directory Servers: Mapping to Attributes" on page 37.

Field	Description	Notes
Data Type	Boolean, String, Integer, Float, Date.	Required. For more information, see "Property Data Types" on page 35.
Description	Text description of the property.	Optional.
External	When checked, Access Manager reads this property value from an external data provider.	Optional.
External Property Provider	The name of the property provider that retrieves the property value from the external data store.	Optional. You cannot view or edit the value of an external property with the Administrative Console. This is because the value of an external property is stored in an external data store.
Read Only	When checked, this field prevents administrators or API programs from modifying the values set for this property.	Optional.
Export/Publish	When checked: • Lets the Runtime API retrieve the values for this property for non-authorization uses, such as personalization. • Makes the property available to HTTP headers.	Optional.
Multi-Value	When checked, if this property is mapped to a multi-value LDAP attribute, there can be more than one value for this property. Properties with Boolean and Date data types cannot store more than one value.	Optional.
Administrative Group	Administrative group that owns the property.	Required. The default is the administrative group of the administrator that created the property.

Field	Description	Notes
Visibility	Identifies the property as public or private. For more information, see "Public and Private Objects" on page 19.	Required.
Help Desk Access	When checked, this field lets Help Desk Admins view the property, regardless of its designation, as private.	Optional.

Properties for Directory Servers: Mapping to Attributes

To add a property in Access Manager based on user information already stored in your directory server, enter a property name in the Administrative Console that is identical to the name of the attribute stored in your directory schema. For more information, see the Help topic "Adding Properties."

Important: If you are using an LDAP data store, after you have mapped an LDAP attribute to an Access Manager property, do not modify the LDAP attribute. If you do so, you cannot save users that reference the associated property. This is not true for Active Directory or SQL.

To create a new property, create identical entries in your directory schema and in the Administrative Console. See your directory server documentation for complete information on creating and modifying attributes using your native directory server administration tools.

Note: To map a property to an LDAP attribute, you must have write permission to the attribute.

For example, to create a property named Age, use your directory server administration tools to create a user attribute named Age in your directory schema. Then, use the Administrative Console to create a property named Age with an integer data type to identify the same value.

Important: Properties must have unique names. They cannot share names with the required user information fields listed in "User Attributes" on page 30. For example, in the default Sun ONE configuration, the following attributes are reserved: cn, givenname, sn, mail, userpassword. Also, avoid system-level attributes. If you are using the default user object class, you cannot use the attributes dn, uid, o and dc as properties.

It is important that you manage properties and their corresponding directory server attributes in parallel. Keep track of both whenever you make changes or additions to your store of properties. However, keep in mind that directory server attributes and Access Manager properties are separate data entities. If you delete a property in the Administrative Console, the corresponding attribute and any values you have set for it still exist in the directory server. If you manually remove an attribute that you have mapped to a property in Access Manager from the directory server, the corresponding property still exists in Access Manager, but is disabled.

For information about mapping LDAP attributes to properties in Access Manager, see the *Servers Installation and Configuration Guide*.

Publishing Properties to the HTTP Header

You can configure Access Manager to automatically publish selected properties to the HTTP header upon successful authentication. This lets you introduce into the session any user information that you define as a property. With this capability, you can use properties to customize or personalize authenticated users.

Associating Properties with Applications

By default, when a Runtime API client requests a property, the Authorization Server exports all properties marked as exportable. To export properties based on the accessed URL, you can associate properties with applications. This ensures that the Authorization Server only exports those properties that are associated with the application being accessed.

For example, suppose you define a URL called marketing/programs.html, which is part of an application called Marketing. Also, suppose you define two exportable properties called Contact and Area, but only associate Contact with the application Marketing. When a client tries to access marketing/programs.html, the Authorization Server only exports Contact.

For instructions on how to associate properties with applications, see the Administrative Console Help topic "Associating Properties with Applications."

Note: In the Administrative Console, when you list all exportable properties associated with an application, the Administrative Console only displays those properties that are currently marked as exportable. If you reconfigure properties, they are no longer exportable, those properties are not included in the list.

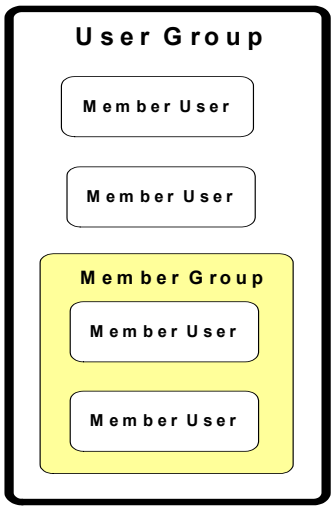
Authentication

You can configure authentication methods in the configuration files of your Agent. For more information, see the RSA Access Manager Agent documentation set.

User Groups

To help you organize access to resources protected by Access Manager, you can create user groups. User groups can include users and other user groups. A user belonging to a user group is called a member user. A user group that belongs to another user group is called a member group. Users can belong to more than one user group. The following figure depicts users and user groups.

Users and User Groups



Note: Special characters cannot be used in user group names. For Active Directory users, the ID can be no longer than 20 characters. If you are using an Active Directory data store, user group names cannot be the same as existing User IDs. If you are using an LDAP data store, you cannot edit the **User Group Name** field.

User groups allow you to use entitlements to grant or deny access to groups of users, rather than to one user at a time. All entitlements granted to a user group automatically apply to all members of the user group. However, individual user privileges take precedence over user group privileges.

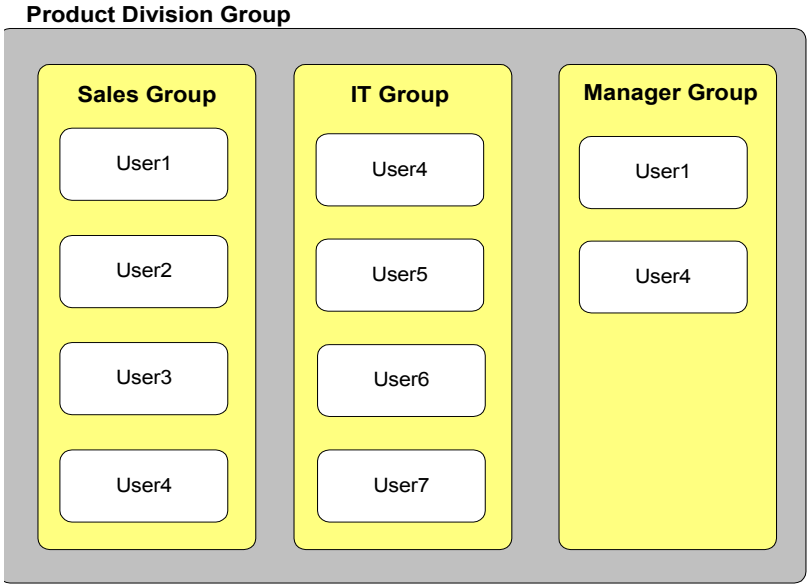
There is no limit on the number of levels of nested groups that you can create. However, the performance penalty increases with each added level of group nesting.

Important: In Active Directory, nested groups are not supported in mixed mode. You can only nest user groups of the same type. As the Administrative Console does not indicate user group type, RSA recommends that you use native Microsoft tools to add member user groups.

You can organize your users into groups to fit the needs of your organization. For example, a school might choose to create two groups:

- A group named Teachers, with access to word processing software, database software, student records, assignments, and test answers
- A group named Students, with access only to software and assignments

Nested groups can be useful for defining logical subgroups, such as divisions within departments, or teams within divisions. In the following figure, the Sales, IT, and Management teams are defined as member groups of the user group, Product Division Group. The following figure depicts a Nested Group.



In an approach similar to traditional access control listing, you can create an entitlement that allows or denies the Sales group access to the IT group's resources, and vice versa. You can control access to system resources by user group membership, as well as by the individual user. For more information about creating entitlements, see Chapter 5, Configuring Security Policies.

A user group must have a unique name and can have an associated description. You manage user groups from the Manage User menu of the Administrative Console.

4 Adding and Managing Resources

Your primary concern as an RSA Access Manager administrator is the protection of system resources. Before you can protect resources, you must add each resource that you want to protect to Access Manager. This chapter describes how to add and manage resources.

Authorization Mode

Before you use the Administrative Console to add resources, decide whether you want to configure your Access Manager environment for active or passive mode. Basic access to resources depends on this configuration. By default, the system is set to behave in passive mode. The two different types of authorization modes are:

- Active Authorization Mode
- Passive Authorization Mode

For more information about configuring the authorization mode, see the *Servers Installation and Configuration Guide*.

Active Authorization Mode

In active authorization mode, all resources are unprotected. The system only protects resources that you explicitly add to the Access Manager system.

Example

The Human Resource (HR) department of a large enterprise implements Access Manager to protect the web servers that contain HR information. Most of this information on the HR web servers must be available to all authenticated clients. All employees must have access to benefits information, the holiday schedule, company events, and so on.

However, the HR web servers also contain pages of sensitive salary and personal information for each employee. This information must be available only to HR managers and the individual employee.

Since the Access Manager administrators want to leave 75% of the web server resources unprotected, they choose to configure the system in active authorization mode. They perform the following administrative tasks necessary to actively protect their sensitive resources:

- Add servers and resources to Access Manager.
- Add user and user group entitlements that explicitly deny access to specific users and user groups.
- Add Smart Rules to allow access based on the value of a user's dynamic attributes and properties, the moment a user attempts to access a resource.

Passive Authorization Mode

In passive authorization mode, all resources are protected. To allow access to a resource, you must add the resource to Access Manager and grant access to users with an entitlement or Smart Rule. This is the default setting.

Example

Many organizations that implement Access Manager have high security needs in which the majority of web-based resources must be closely protected. For example, an online stock trading site might maintain only a few directories of public material freely available to all users. All other information regarding user portfolios and accounts is highly sensitive.

Even within the directories of public material, there is only a small portion of the information that the company wants to make available to every user. For example, prospective clients must be able to view the home page and demo pages, but analysis of stocks and mutual fund information must be available only to members.

Since the administrators want to leave only 5% of the web-based resources unprotected, they choose to configure the system in passive authorization mode. They then perform the following administrative tasks necessary to passively protect their sensitive resources:

- Add servers and resources to Access Manager.
- Add user and user group entitlements that explicitly grant or deny access to specific users and user groups.
- Add Smart Rules to allow access based on the value of a user's dynamic attributes and properties, the moment a user attempts to access a resource.

Servers

Web servers and application servers host the resources you protect with Access Manager. Before you can protect an application or resource, you must identify to Access Manager the server where the resource resides.The following table describes the different server fields.

Server Information	Description	Comments
Server Name	The name by which the server is identified to Access Manager.	Required. Can be from 1 to 255 characters. Each server in the system must have a unique name. Server names must match the name assigned in the Agent configuration parameter.
Server Type	The type of server you want to add.	Required. This is either a web server, an application server, or an enhanced application server.
Product	The manufacturer's product name for the server.	Required. For example, Apache, Microsoft, or BEA WebLogic.
Hostname	This must match the actual, fully-qualified name of the server.	Required. For example, hostname.domain.com, or the server IP address. Can be from 1 to 255 characters.
Port	This is the port address on which the server advertises its HTTP services.	Required. The default is port 80. Valid range is 1 to 65535.
Description	Text description of the server.	Optional.
Administrative Group	The administrative group that owns this server.	Required.
Visibility	Identifies the server as either public or private.	Required.

For instructions on adding servers, see the Administrative Console Help topic "Adding Servers."

Web Servers

Web servers host web resources. Resources can be directories or files such as web pages or graphic files. To protect web servers, they must be identified to Access Manager by name, not by domain name, and they must have an Agent installed and running.

When you add a web server to Access Manager, you must define basic information. For details about the information you enter when you add a server, see the preceding section, Servers.

For information about adding resources to web servers, see the Administrative Console Help topic "Understanding Resources."

Application Servers and Enhanced Application Servers

Access Manager divides application servers into two categories:

Application Servers. Protection of resources is configured partially by the Agent and partially by the Administrative Console.

Enhanced Application Servers. Protection of resources is configured entirely by the Administrative Console.

To protect resources hosted on any application server, you must first define the server by naming it and providing basic information. For details about the information you enter when you add a server, see "Servers" on page 43.

For information about adding J2EE web or EJB resources to application servers, see "Defining J2EE Resources on Application Servers" on page 51.

For information about adding J2EE resources to enhanced application servers, see "Defining J2EE Resources on Enhanced Application Servers" on page 47.

Mirror Sites

If an organization has several load-balanced servers acting as mirror sites, that is, they all serve the same content and have the same directory structure and files then these servers can share the same name in the Administrative Console.

Applications

An application is a collection of resources in the Access Manager system that are logically grouped together and named. The grouping of resources into applications lets you apply a security policy to related resources of different types. Resources included in an application can include URLs for web pages, CGI files, directories, GIF or JPG files, and functions.

For more information about applications, see the Administrative Console Help topic "Understanding Applications."

Resources

You identify resources to Access Manager by adding them to applications. You can create HTTP web resources, as well as the following J2EE resources: EJBs, JMS, JDBC, JNDI, EIS, and J2EE web resources. After you add resources, you can create security policies with Smart Rules and entitlements that grant or deny access to the resources.

For more information on adding resources, see the Administrative Console Help topic "Adding Resources."

Wildcard Characters

Access Manager supports the use of wildcard characters in file specifications for resources, such as *.jpg or index.*, and for resources that have a wildcard character as the last character in the string, such as /hr/benefits/*. You cannot use embedded wildcard characters, such as /marketing/*/sample.html.

Wildcard Processing

When a user attempts to access a protected URL, the Authorization Server checks for a resource that exactly matches the requested URL, and then determines whether the user has permissions to access the URL. If the Authorization Server does not find a resource that exactly matches the URL, it checks for resources that are less specific matches, for example, a URL that ends with the same file name, but of a different file type.

If the Authorization Server finds multiple matches, it only checks the permissions on the match closest to the requested URL. If the user does not have permissions for the best match, the user is denied access regardless of the permissions for any other matches.

For example, if a user requests the URL, www.<*example.com/index.html*>, the Authorization Server looks for permissions on resources in the following order:

1. A file with the exact name and file type the user requested, for example, /index.html
2. A file of any file type that has the exact name that the user requested, for example, /index.*
3. A file of any name that has the same file type of the file requested, for example, /*.html
4. A file of any name and any file type, for example, /*.*
5. A directory of any name, for example, /*

Access Manager evaluates the first match it finds and ignores any other matches.

Example 1: Suppose /index.* and /*.html are both resources on your web server. You have an entitlement that allows you access to /index.* and an entitlement that denies you access to /*.html.

If you try to access /index.html on your web server, you are allowed access because the first resource to match is /index.* and there is an allow permission on /index.*. However, if you have an entitlement that denies you access to /index.*, you are denied access.

Example 2: Suppose /index.html and /*.html are both resources on your web server. You have no entitlements for /index.html, and an entitlement that grants you access to /*.html. If you try to access /index.html on your web server, you are denied access because the first resource to match is /index.html and you have no entitlements for that specific resource even though you have access to /*.html.

Important: If you define a resource as `*.html`, entitlements or Smart Rules for that resource apply to any URL that ends in .html. Assess the impact of this change before you upgrade your system from a version previous to RSA ClearTrust 5.5.3 or later.

Defining URLs as Resources

When entering URLs as resources in the Administrative Console, follow these basic rules for URL syntax:

- Begin all URL definitions with "/"
- Enter fully-specified URLs, or end URLs with "/*"
- To protect an entire directory, end the URL definition with "/*"

Important: When you protect an entire web server, use "/*" sparingly. When you use "/*" to protect the entire web server, you may block access to graphics and objects associated with logon or self-registration forms.

Incorrect URL syntax can cause security holes. For example, if you want to protect all resources under the Finance_Server/Projections directory, and define the URL in this way:

(error example -- do not define directories using this syntax)

/Finance_Server/Projections

Unauthorized users can still gain access to this URL with a trailing slash:

/Finance_Server/Projections/

The directory is only securely protected if you define the URL with "/*" at the end:

/Finance_Server/Projections/*

Limitations on Resources

Resources are subject to the following limitations:

- A URL cannot be added to more than one application simultaneously.
- As a URL can be a directory, URLs in different applications can overlap.
- A single application can contain URLs on more than one server, as long as the administrative group that owns the application also owns all the servers.

URLs in Overlapping Applications

A URL can only exist in one application, but because URLs can contain wildcard characters, it is possible for multiple URLs in different applications to influence whether or not a user can access a single resource.

For example, in Application A, suppose a wildcard is used in a URL to protect the entire finance web server. The URL is finance/*. Application B protects only the URL finance/salary_info.html. Now suppose an entitlement grants a user access to Application A, while a Smart Rule denies the user access to Application B. The URLs in these two applications overlap and conflict. In a situation like this, Access Manager applies the rule of "most specific rule wins," which in this example is finance/salary_info.html. As a result, the user is allowed to access everything in the web server finance, except for salary_info.html.

Defining J2EE Resources on Enhanced Application Servers

Access Manager supports the following types of J2EE resources:

- Enterprise Java Beans (EJB)
- Java Messaging Service (JMS)
- Java Naming Directory Interface (JNDI)
- Java Database Connection (JDBC)
- Enterprise Information System (EIS)
- Web (JSP and HTML pages, servlets, and so on)

Enterprise Java Bean Resources (EJB)

An EJB resource specifies access to all methods or specific methods of an Enterprise Java Bean. The following table lists the different EJB fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
Deployed Application Name	Yes	The name under which the application is deployed on the application server.	Cannot use / or *.

Field Name	Required	Possible Values	Validation
Module	Yes	The name of the module in which the resource is located.	Cannot use / or *.
EJB Name	Yes	The name of the EJB you want to protect.	Cannot use / or *.
Method Interface	Yes	Remote, Home, Local, LocalHome.	
Method	Yes	Can be a wildcard character or a method name.	Cannot use /.
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

Java Messaging Service Resources (JMS)

A JMS resource specifies access to the Java Messaging Service, which uses queues or topics for communications between applications. The following table lists the different JMS fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
JMS Resource	Yes	The name of the JMS resource you want to protect. Can be a wildcard character or a name.	Cannot use /.
Messaging Type	Yes	Topic or Queue.	
Action	Yes	Send, Receive, or Both. If the **JMS Resource** field is a wildcard character, the **Action** field must be Both.	
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

Java Naming Directory Interface Resources (JNDI)

A JNDI resource specifies access to the Java Naming Directory Interface service, which is used as a large directory to all the applications known to the server. The following table lists the different JNDI fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
JNDI Resource	Yes	The name of the JNDI resource you want to protect.	Cannot use /.
Action	Yes	Lookup, Modify, List, All Actions.	
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

Java Database Connection Resources (JDBC)

A JDBC resource specifies access to the Java Database Connection through the pool to which the resource is connecting. The following table lists the different JDBC fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
JDBC Resource	Yes	The name of the JDBC Resource you want to protect. Can be a wildcard character or a resource name.	Cannot use /.
Pool Type	Yes	Connection Pool or Multi-Pool.	
Action	Yes	Reserve, Admin, Shrink, All Actions. If the **JDBC Resource** field is a wildcard character, the **Action** field must be All.	
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

Enterprise Information System (EIS)

An EIS resource represents the data coming from a legacy database or any Enterprise Information System through a Java Connectivity Architecture (JCA) connection. The following table lists the different EIS fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
Deployed Application Name	Yes	The name under which the application is deployed on the application server.	Cannot use / or *.
Module	No	The name of the module in which the resource is located.	Cannot use / or *.
EIS Resource	Yes	The name of the EIS Resource you want to protect.	Cannot use /.
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

J2EE Web Resources

A J2EE web resource is a web-based resource stored on an application server, such as JSP pages, HTML pages, image files, and servlets. The following table lists the different Web Resource fields.

Field Name	Required	Possible Values	Validation
Enhanced Application Server	Yes	The name of the application server where the resource resides.	
Web Resource	Yes	The URL of the resource you want to protect. This can be a wildcard character, or a specific URL, such as a JSP or HTML file.	Allows /, *, and //. The wildcard must come last in the string, for example, marketing/*.
Method	No	Get, Post, Put, Delete, Options, Trace, All.	
Policy Conflict Resolution	Yes	Allow access when policy conflicts occur or deny access when policy conflicts occur.	

Defining J2EE Resources on Application Servers

For instructions to define J2EE resources on your Application Server, refer to your Application Server documentation.

Functions

A function is an Access Manager representation of any type of function or method in any type of custom application that you build. Modeling a method as a function allows Access Manager to control access to that method. This allows you to implement Access Manager Agent-like controls (similar to building an Access Manager WAX) governing access to methods in your custom applications.

Policy Conflict Resolution

For each resource you create in Access Manager, you set the policy conflict resolution setting, either to Allow access when policy conflict occurs or Deny access when policy conflict occurs. You can change the setting any time you add or edit applications or resources.

The policy conflict resolution setting becomes important when the system makes access control decisions. If one policy allows access and another denies access to a given resource (and the policies are of equivalent specificity), the system checks your policy conflict resolution setting to decide which policy takes priority.

Therefore, when setting the policy conflict resolution setting for a given resource, you must consider the sensitivity of the resource and the manner in which you want to resolve conflicting access policies to the resource. To deny access when policies conflict, set the policy conflict resolution setting for each resource to Deny access in the case of a conflict. To allow access when policies conflict, set the policy conflict resolution setting for each resource to Allow access in case of conflict.

Note: The policy conflict resolution setting affects system access control decisions, when entitlements conflict or there are multiple Smart Rules on a given resource. It is not a global setting governing all attempts to gain access to the resource. The setting is important only when there are conflicting security policies as described in "Resolving Multiple or Conflicting Entitlements" on page 54.

The scenarios for processing conflicting entitlements or multiple Smart Rules can become complex, but the policy enforcement results can be accurately summarized in their priority ordering. Following is the order of priority for applications and resources:

1. User entitlements on the URL or function
2. Group entitlements on the URL or function
3. Smart Rules on the URL or function

4. User entitlements on the application that includes the URL or function
5. Group entitlements on the application that includes the URL or function
6. Smart Rules on the application that includes the URL or function

5 Configuring Security Policies

After configuring delegated administration, you determine users and user groups, and identify the resources you want to protect. You must decide how you want to grant access to resources. You control access using a security policy.

After a security policy is established for a resource, users with the required credentials are given access to the protected resource. When a user lacks the required credentials, the system displays an access denied error message. Access Manager can also provide the reason why the user was denied access to a resource, this helps to develop a custom application that can automatically help the user obtain the required credentials.

This chapter describes the two types of Access Manager security policies:

- Entitlements
- Smart Rules

Entitlements

You use entitlements to allow or deny a user or user groups access to a resource. Entitlements are the most specific type of security policies. They always take precedence over Smart Rules. Use entitlements when you want to:

- Explicitly grant a user or user group access to a resource.
- Explicitly deny a user or user group access to a resource.

For example:

- If a user must always be allowed access to a protected resource, such as a web server, create an entitlement that grants the user access to the web server. After you create the entitlement, the user is always allowed access to the web server.
- If a user must never be allowed access to a protected resource, such as an application server, create an entitlement that denies the user access to the application server. After you create the entitlement, the user is always denied access to the application server.

You create entitlements in the Administrative Console. You can specify two types of entitlements:

User entitlements. Entitlements assigned at the user level affecting only one specific user.

User group entitlements. Entitlements assigned at the user group level affecting all member users and member groups in the user group.

When users are denied access to a resource, the system returns the access denied error message:

```
You are not authorized to access this page
```

For information about adding entitlements, see the Administrative Console Help topics "Adding User Entitlements" and "Adding User Group Entitlements."

Entitlement Precedence

If you allow or deny a user group to access a resource through a user group entitlement, all members of the user group are allowed or denied access to the resource, except when a contradicting user entitlement exists. A user entitlement, which applies to a specific user, is more specific than a user group entitlement, which applies to a group of users. Therefore, the user entitlement always takes precedence over a user group entitlement.

Example

If an entitlement allows the user group West Coast Users access to the resource shipping/index.html, all members of West Coast Users are allowed access to shipping/index.html. Though Chuck Jackson is a member of West Coast Users, an entitlement specifically denies him access to shipping/index.html. In this case, all members of West Coast Users are granted access to shipping/index.html, except for Chuck Jackson.

Resolving Multiple or Conflicting Entitlements

The entitlements can exist at different levels and it is possible to create conflicting entitlements. When entitlements conflict, these are the criteria for resolving conflicts, in order of priority:

1. Specificity of the resource
2. Specificity of the entity
3. Policy conflict resolution setting

The fundamental principle is "most specific match wins," and when specificity is equal, the policy conflict resolution setting for the resource determines the resolution of conflicting entitlements.

Specificity of the Resource

Access Manager applies a "most specific match wins" logic to resources. This means that security policies based on resources and functions take priority over security policies based on the applications that include them.

For example, suppose you have a URL, marketing/proposals/index.html, which belongs to an application, which contains multiple URLs. A security policy based on the URL, marketing/proposals/index.html, takes precedence over the application to which this URL belongs.

The system also evaluates the specificity of the URL defining a resource. If a user has entitlements at various levels in a web server directory, the most specific entitlement determines whether the user is granted access.

For example, Entitlement I grants user group access to the Profits directory while Entitlement II denies access to the more specific Executive directory:

Entitlement I: Allow the group "Junior Analysts" access to Finance_Server/Projections/Profits/*

Entitlement II: Deny the group "Junior Analysts" access to Finance_Server/Projections/Profits/Executive/ Q2_Exec_Summary.html

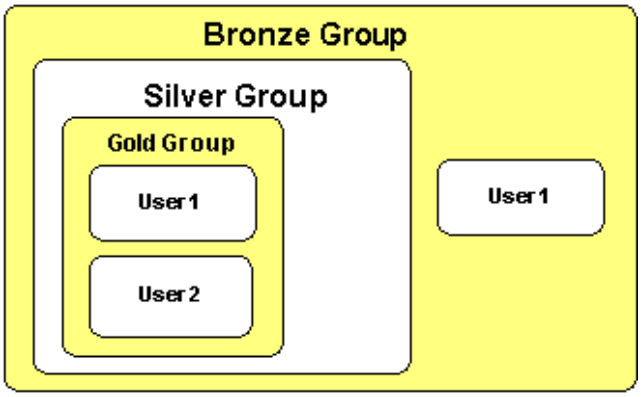
The Entitlement II takes precedence over Entitlement I. This is because the Entitlement II is more specific. These rules allow the group full access to all resources under Profits, except the more specifically defined Executive resources.

Specificity of the Entity

Access Manager applies a "most specific match wins" processing logic to entities. This means:

- User entitlements take priority over user group entitlements.
- Member user group entitlements take priority over entitlements on user groups that include them.

In the following figure, Users 1 and 2 belong to the Gold group and, by nested group membership, also belong to the Silver and Bronze groups.



Example 1: User 1 and its parent group have conflicting entitlements.

- Entitlement 1: Deny User 1 access to index.html
- Entitlement 2: Allow Gold Group access to index.html

User-level specificity wins over group-level specificity. Therefore, attempts to gain access produce the following results:

- User 1 is denied access based on specific user entitlement.
- User 2 is allowed access based on group allow rule.

Example 2: Gold and its parent group Silver have conflicting entitlements.

- Entitlement 1: Deny Silver Group access to index.html
- Entitlement 2: Allow Gold Group access to index.html

Member group-level specificity wins over parent group-level specificity. Therefore, when User 2 attempts to gain access, it is allowed access because:

- Gold Group is more specific than its parent Silver.
- User 2 has no specific deny rule at the user level.

Example 3: Bronze and its member group, Gold, have conflicting entitlements.

- Entitlement 1: Deny Bronze Group access to index.html
- Entitlement 2: Allow Gold Group access to index.html

This must be resolved by the policy conflict resolution setting. Therefore, attempts to gain access produce the following results:

- User 2 is allowed access because Gold group is more specific than Bronze group.
- User 1, however, is a member of both groups. Therefore, the groups are of equal specificity to the user. In this case, the system considers the policy conflict resolution setting specified for the resource index.html.

Policy Conflict Resolution

Policies that allow access, or policies that deny access, take priority depending on the policy conflict resolution setting of the resource in question.

For more information, see "Policy Conflict Resolution" on page 51.

Smart Rules

You use Smart Rules to allow or deny a user access to a resource based on the value of a user property at the moment the user attempts to access the resource. You can apply Smart Rules to any resource.

When a user tries to access a protected resource, Access Manager checks the user properties associated with the Smart Rules protecting the resource and grants or denies access based on the criterion of each Smart Rule.

Important: Smart Rules decide a user's access to a specified application only if no relevant entitlement exists at any level (user or group). Entitlements always take precedence over Smart Rules.

Each Smart Rule compares a specific user property value to an administrator-specified comparison criterion according to one of the comparison operators in the following table.

Data Type	Operator
Date	Before, After, Is Equal
Boolean	Is

Data Type	Operator
String	Starts With, Contains, Does Not Contain, Ends With, Is Equal To, Is Greater Than, Is Greater Than Or Equal To, Is Less Than, Is Less Than Or Equal To, Is Not Equal To
Integer, Float	>=, <, =, >, <=, !=

For example, an online banking company can create a Smart Rule that allows only users with an account balance of more than $500.00 to access a certain page on their web site. This Smart Rule would take the following form:

Allow if Account Balance > 500

In this case, the user property value is "Account Balance." The comparison criterion is "500." The comparison operator is "greater than" (>).

If users have an account balance of more than $500.00, they are allowed access to the page. If users have an account balance of less than $500.00, they are denied access to the page. When a user is denied access to a resource, the system displays an access denied error message.

For information about creating a Smart Rule, see the Administrative Console Help topic "Adding Smart Rules."

Types of Smart Rules

There are three types of Smart Rules: Allow, Deny, and Require. These three types can be combined in various ways to implement business rules in controlling access to a resource.

If multiple Smart Rules protecting the same resource contradict each other, a policy conflict occurs. For information about how Access Manager resolves policy conflicts, see "Policy Conflict Resolution" on page 51.

Allow Rules

If a user property matches an Allow rule, the user is allowed access. For example, Resource A is protected by this Smart Rule: Allow if State Equals CA.

User A has the State property set with the value "CA." User B has the State property set with the value "WA."

User A's state property matches this Allow rule, so User A is allowed access. User B's state property does not match this Allow rule, so User B is not allowed access to Resource A.

If a resource is protected by several Allow rules, access is allowed if a user property matches any rule.

Deny Rules

If a user property matches a Deny rule, the user is denied access. For example, Resource A is protected by this Smart Rule: Deny if Age < 21.

User A has the Age property set with the value "18." User B has The Age property set with the value "30."

User A's age property matches the Deny rule, so User A is denied access to Resource A. User B's age property does not match the deny rule, so User B is allowed access to Resource A.

If a resource is protected by several Deny rules, access is denied if a user property matches any rule.

Require Rules

If a user property matches a Require rule, the user is allowed access. For example, Resource A is protected by this Smart Rule: Require Account > 500.

User A has the Account property set with the value "600." User B has the Acccount property set with the value "400."

User A's account property matches the Require rule, so User A is given access to Resource A. User B's account property does not match the Require rule, so User B is not allowed access to Resource A.

If a resource is protected by several Require rules, access is granted only if a user's properties match all the rules.

Deny Rules and Authorization Modes

If a user has a property with a value "N/A" (not yet entered), a Deny rule based on that property does not deny access for that user. If that user's access status is not established by a higher level entitlement for that resource, access status is determined by one of these settings:

- The Authorization Server set to active mode allows access.
- The Authorization Server set to passive mode denies access.

For more information about Deny rules, see "Deny Rules" on page 57.

For more information about active and passive modes, see "Authorization Mode" on page 41.

Combining Smart Rules

Allow rules and Deny rules can be combined with Require rules to implement business rules when controlling access to a resource. If Smart Rules of different kinds protect a resource, the order in which they are evaluated is governed by the policy conflict resolution setting applied to the resource. For more information, see "Policy Conflict Resolution" on page 51.

The policy resolution setting is selected during the process of assigning a resource to be protected by Access Manager. When the default conflict resolution setting, "Allow access when policy conflicts occur," is selected, Allow rules are evaluated first, then Deny rules, and finally Require rules. When the alternative setting, "Deny access if policy conflicts occur," is selected, Deny rules are evaluated first, then Allow rules, and finally Require rules. The Smart Rules are evaluated until a user is either denied or allowed access.

The following examples show how four users with different properties are evaluated when trying to access a resource protected by an Allow, a Deny, and a Require rule.

Example 1

The following table shows the Smart Rules protecting Resource A and the properties of User A.

Smart Rules Protecting Resource A	User A Properties
Allow if State Equals CA	State with the value TX
Deny if Age < 21	Age with the value 23
Require Valid Credit Card Is True	Valid Credit Card with the value True

If the policy conflict resolution for this resource is set to "Allow access when policy conflicts occur," the Smart Rules are evaluated in this order:

1. Allow if State Equals CA
2. Deny if Age < 21
3. Require Valid Credit Card Is True

In this case, User A's State property (TX) does not match the "Allow if State Equals CA" rule, so User A is denied access to Resource A. The remaining Smart Rules are not evaluated because the Allow rule has denied access. By eliminating unnecessary evaluation of rules, system performance is improved.

Example 2

The following table shows the same Smart Rules protecting Resource A and the properties of User B.

Smart Rules Protecting Resource A	User B Properties
Allow if State Equals CA	State with the value CA
Deny if Age < 21	Age with the value 18
Require Valid Credit Card Is True	Valid Credit Card with the value True

If the Policy Conflict Resolution for this resource is set to "Deny access when policy conflicts occur," the Smart Rules are evaluated in this order:

1. Deny if Age < 21
2. Allow if State Equals CA
3. Require Valid Credit Card Is True

In this case, User B's Age property (18) matches the "Deny if Age < 21" rule, so User B is denied access to Resource A. The remaining Smart Rules are not evaluated, because the Deny rule has denied access.

Example 3

The following table shows the same Smart Rules protecting Resource A and the properties of User C.

Smart Rules Protecting Resource A	User C Properties
Allow if State Equals CA	State with the value CA
Deny if Age < 21	Age with the value 23
Require Valid Credit Card Is True	Valid Credit Card with the value False

If the Policy Conflict Resolution for this resource is set to "Deny access when policy conflicts occur," the Smart Rules are evaluated in this order:

1. Deny if Age < 21
2. Allow if State Equals CA
3. Require Valid Credit Card Is True

In this case, User C does not match the "Deny if Age < 21" rule, matches the "Allow if State Equals CA" rule, but does not match the "Require Valid Credit Card Is True" rule, so User C is denied access to Resource A.

Example 4

The following table shows the same Smart Rules protecting Resource A and the properties of User D.

Smart Rules Protecting Resource A	User D Properties
Allow if State Equals CA	State with the value CA
Deny if Age < 21	Age with the value 23
Require Valid Credit Card Is True	Valid Credit Card with the value True

If the Policy Conflict Resolution for this resource is set to "Deny access when policy conflicts occur," the Smart Rules are evaluated in this order:

1. Deny if Age < 21
2. Allow if State Equals CA
3. Require Valid Credit Card Is True

In this case, User D does not match the "Deny if Age < 21" rule, matches the "Allow if State Equals CA" rule, and matches the "Require Valid Credit Card Is True" rule, so User D is given access to Resource A.

Evaluating Smart Rules in Sequential Order

Access Manager can be configured to ignore policy conflict resolution settings and evaluate the Smart Rules in the order they appear in the Administrative Console.

By configuring Access Manager to ignore policy conflict resolution settings, it is possible to arrange different types of Smart Rules in an arbitrary list. The following table shows a group of Smart Rules that control access to the wine section of an online supermarket as they appear in the Administrative Console:

Smart Rule	Access
Age >= 21	Require
Valid Credit Card Is True	Require
Encryption Off Is True	Deny
Bad Credit Is False	Require
Account Closed Is True	Deny
Valid Username Is True	Allow
Valid PIN Is True	Allow

The "Age >= 21" Require rule must be matched in order to shop in the wine section, because it is the most important rule in this group. It is placed at the top of the list so that the other Smart Rules do not need to be evaluated if the first Rule is not matched. Reducing unnecessary evaluation of rules improves performance.

Smart Rules and User Properties with Multiple Values

Smart Rules can include user properties that have more than one value.

For example, a resource is protected by this Smart Rule: Allow if Department Equals Sales.

User A works in more than one department and has the Department property set with the values "Marketing," "Sales." User B also works in more than one department and has the Department property set with the values "Marketing," "Customer Support."

In this case, User A is allowed access to this resource because the Allow rule is matched (User A is in Sales). User B, however, is denied access because the Allow rule is not matched (User B is not in Sales).

Smart Rules with multi-value user properties that use the operators "Does Not Contain" or "Does Not Equal" are only satisfied if none of the property values match the comparison criterion. For all other operators, only one of the property values needs to match the comparison criterion to satisfy the Smart Rule.

Smart Rule Examples

These additional examples further illustrate the use of Smart Rules.

Example One

An insurance company using Access Manager has customers throughout the south and northwest United States. At the beginning of its fiscal year, the company decided to make a special offer available to residents of California, Texas, and Oregon on its web site. To accomplish this, the company created three Smart Rules to control access to the page on the web server containing the special offer. The company had already created a property called State, which is normally used as part of the customer's mailing address:

- Allow if State Equals CA
- Allow if State Equals TX
- Allow if State Equals OR

This simple setup accomplished the desired goal. Residents of California, Texas, and Oregon can access the special offer, but everyone else is denied access.

A month later, the insurance company decided to limit the offer to users with good credit ratings. Since there is already another property called Bad-Credit, (a Boolean that is set to true if the account has been flagged for non-payment), adding another Smart Rule is straightforward:

- Deny if Bad Credit Is True
- Allow if State Equals CA
- Allow if State Equals TX
- Allow if State Equals OR

Deny rules are evaluated first because the Policy Conflict Resolution setting for this resource is set to **"**Deny access when policy conflicts occur". Only users with good credit from California, Texas, or Oregon can access the insurance company's special offer web page.

Example Two

It is also possible to combine Require rules. In this example, a company wants to limit access to an area of its web site to retail customers that have account balances over $100. In this case, both parts of the condition must be met or the user is denied access. The Access Manager Administrator creates two Smart Rules:

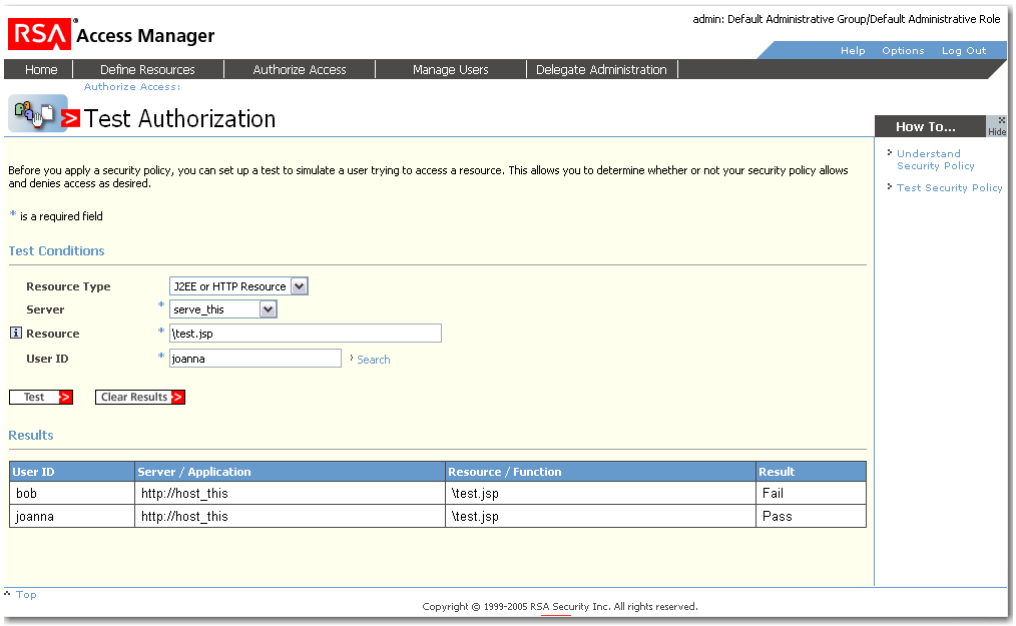
- Require Account Balance > 100
- Require Account Type Equals Retail

At runtime, only Retail users with account balances in excess of $100 are allowed access to the site.

Testing Security Policies

Before you apply a security policy, you can use the testing tool in the Administrative Console to simulate a specific user's attempts to access a specific resource. This allows you to determine whether your security policy allows and denies access as you intended.

In the following figure, the user "bob" is denied access to a resource by an entitlement. The user "joanna," however, is granted access to this page by an entitlement. Joanna's test passes.



For more information, see the Administrative Console Help topic "Testing Security Policy."

Index

S

T

U

W