Oracle® COREid
Access and Identity

Administration Guide

Volume 2: Access System Administration

10*g* Release 2 (10.1.2)
Part No. B19008-01

May 2005

ORACLE

This program contains third-party code from Apache. Under the terms of the Apache Software License, Oracle is required to provide the following notices. Note, however, that the Oracle program license that accompanied this product determines your right to use the Oracle program, including the Apache software, and the terms contained in the following notices do not change those rights. Notwithstanding anything to the contrary in the Oracle program license, the Apache software is provided by Oracle "AS IS" and without warranty or support of any kind from Oracle or Apache.

Contents

Preface

The Administration Guide is divided into two volumes. This book, *Volume 2*, provides information on configuring COREid to define access controls, using COREid to control user access to applications and data, and configuring single sign-on.

Note: Oracle *COREid* was previously known as Oblix *Netpoint*. All legacy references to Oblix and NetPoint, for example, in screen shots, illustrations, and documentation titles, should be understood to refer to Oracle and COREid, respectively.

This Preface covers the following topics:

Intended Audience

This guide is intended for COREid administrators, Master Access Administrators and Delegated Access Administrators. Administrators configure the rights and tasks available to other administrators and end users. A COREid Administrator, the highest level administrator, is selected during COREid installation and setup. The COREid Administrator delegates responsibilities to other administrators.

This document assumes that you are familiar with your LDAP directory and Web servers and with *Volume 1* of this guide.

COREid Documentation

The manuals that are available for this release include:

Introduction to COREid—Provides an introduction to COREid, a road map to COREid manuals, and a COREid glossary of terms.

COREid Release Notes—Provides up-to-the minute details about the latest COREid release.

COREid Installation Guide—Explains how to install and configure the COREid components.

COREid Upgrade Guide—Explains how to upgrade earlier versions of COREid to the latest version of COREid.

COREid Administration Guide—Explains how to configure COREid applications to display information stored in the directory, how to assign view and modify permissions for data displayed on the COREid applications, and how to assign access controls to users.

COREid Deployment Guide—Provides information for people who plan and manage the environment in which COREid runs. This guide covers capacity planning, system tuning, failover, load balancing, caching, and migration planning.

COREid Customization Guide—Explains how to change the appearance of COREid applications and how to control COREid by making changes to operating systems, Web servers, directory servers, directory content, or by connecting CGI files or JavaScripts to COREid screens. This guide also describes the Access Server API and the Authorization and Authentication Plug-in APIs.

COREid Developer Guide—Explains how to create AccessGates and how to develop plug-ins. This guide also provides information to be aware of when creating CGI files or JavaScripts for COREid.

COREid Integration Guide—Explains how to set up COREid to run with third-party products such as BEA WebLogic, the Plumtree portal, and IBM Websphere.

COREid Schema Description—Provides details about the COREid schema.

Online Help is available from each COREid screen.

Typographical Conventions

COREid manuals use the following typographical conventions:

- When you are instructed to select elements sequentially, the actions are separated with angle brackets, as shown below:

  Click System Admin > System Configuration > View Server Settings.

- Paths to a file are shown using syntax for either the Unix or Windows platform:

  /COREid_install_dir/identity/oblix/logs/debugfile.lst
  \COREid_install_dir\identity\oblix\logs\debugfile.lst

  where *COREid_install_dir* refers to the directory where the component, in this case, the COREid Server, is installed.

Contact Information

For a list of contacts including corporate offices world wide, sales, and other details, visit the Oracle Web site at:

`http://www.oracle.com`

You can contact Oracle with questions or comments as follows:

Customer Care—http://www.oracle.com/support/contact.html

Corporate Headquarters

Oracle maintains offices world wide. Oracle corporate headquarters is located at:

500 Oracle Parkway
Redwood Shores, CA 94065
Phone: (650) 506-7000

Before Contacting Customer Care

Before contacting Customer Care, please have available the following:

- Oracle product name and version number
- Type of computer and operating system you are using

Accessing the Customer Care Knowledge Base

For more information about using COREid, see the Oracle Customer Care Knowledge Base. To access the Knowledge Base, you need a login name and password, which you can obtain from your Oracle sales representative.

To access the Knowledge Base:

1. Enter the following URL in your browser and press Return.

   ```
   http://www.oracle.com/support/contact.html
   ```

2. Click the phrase, Login to the Oracle PremiumCare Online Portal.
3. Enter your user name and password in the box that appears, then click Login.
4. Under Oracle Support Tools, click Case Manager.
5. In the next screen, click Find Answers to gain access to the Knowledge Base.

Section I: Configuring the Access System

1 Configuring Access Administrators and Server Settings

This chapter explains how to assign Access System administrators, install a permanent license from Oblix, and manage other server settings. Included here are the following topics:

For more information about managing the Access System, see:

Prerequisites

NetPoint 7.0 should be installed and setup, as described in the *NetPoint 7.0 Installation Guide*. Read the *Introduction to NetPoint 7.0* manual, which provides an overview of NetPoint not found in other manuals. Also, familiarize yourself with *Volume 1*, which provides a brief review of Access System applications and installation; introduces Access System configuration and administration; and includes common functions, configuration, and administration.

Configuring Access Administrators

The Access System enables the protection of online resources by enforcing policy-based authentication and authorization rules. The Access System also enables Web single sign-on.

In addition to the NetPoint Administrator, there are two types of NetPoint *Access* Administrators who can configure and manage the Access System:

- **Master Access Administrators**—These administrators have the right to perform any task in the Access System except the right to create other Master Access Administrators.
- **Delegated Access Administrators**—These administrators only have the right to perform tasks that a Master Access Administrator delegates to them.

The following table summarizes the privileges of these types of administrators. Master Access Administrators automatically have these privileges while Delegated Access Administrators must be explicitly granted these privileges:

Privilege	Description	Who Performs This Task
Generate a shared secret	Create a cryptographic key that encrypts single sign-on cookies. See “Creating a Shared Secret Key” on page 328.	Master Access Administrator
Configure the Master Audit Rule	The Access System will not log any audit information to the audit log file until a Master Audit Rule exists. See “Configuring the Master Audit Rule” on page 132. For more information about logging, see *Volume 1*.	Master Access Administrator
Flush the password policy cache	See “Flushing Password Policy Caches” on page 330.	Master Access Administrator
Configure ReadyRealm	Provide way for BEA WebLogic customers to use NetPoint to control user access and manage identities for their applications. See the *NetPoint Integration Guide* for details.	Master Access Administrator
Manage AccessGates	View, create, and configure one or more instances of an AccessGate. See “Configuring AccessGates” on page 52.	Master *and* Delegated Administrator
Manage Access Servers	Configure an Access Server to communicate with AccessGates and a directory server. See “Configuring Access Servers” on page 35.	Master *and* Delegated
Manage Access Server clusters	See “Managing Access Server Clusters” on page 45.	Master *and* Delegated

Privilege	Description	Who Performs This Task
Manage Authentication Schemes	Authentication is the process of proving that a user is who he or she claims to be. See “Configuring User Authentication” on page 149.	Master *and* Delegated
Manage Authorization Schemes	Authorization is the process of determining if a user has the right to access a requested resource. See “Configuring User Authorization” on page 229.	Master *and* Delegated
Manage Host Identifiers	Identify the names by which users can identify a host. See “Using Preferred Hosts or Host Identifiers” on page 79.	Master *and* Delegated
Manage Resource Type definitions	Define the kind of resource to be protected, including its associated operations. See “Resource Types Defined by NetPoint” on page 108.	Master *and* Delegated
Manage User Configuration	Create and modify a list of users who are prohibited from accessing any of your resources and flush these users from the cache. See “Access System Configuration and Management” on page 319.	Master *and* Delegated

The following sections describe how to configure these administrators and delegate administrative tasks. You complete these tasks using the Access System Console, System Configuration function.

Note: The delegation of administrative responsibilities for a policy domains works somewhat differently from the delegation of other responsibilities. See “Delegating Policy Domain Administration” on page 145 for details.

Configuring Master Access Administrators

Only NetPoint Administrators can create Master Access Administrators. A Master Access Administrator can perform any function in the Access System except for creating other Master Access Administrators, and can delegate administrative functions.

Note: You must be a Master Access Administrator to create a shared secret key that encrypts single sign-on cookies sent from an AccessGate to a browser. You should generate a cryptographic key as soon as possible after installing NetPoint, otherwise a less secure default is used. See “Creating a Shared Secret Key” on page 328.

To add a Master Access Administrator

1. From the Access System Console, select System Configuration > Configure Admins.

   The Configure Administrators page lists current Master Access Administrators.

2. Click Master Access Administrators.

   The Modify Master Access Administrators page appears.

3. Click Select User.

   The Selector page appears.

4. Use the Selector to select the persons you want.

5. Click Done to return to the Modify Master Access Administrators page.

   The names of any new people you chose using the Selector are displayed in the Modify Master Access Administrators page.

6. Use the checkboxes to deselect any names that you need to remove from your list.

7. Review your selections to ensure that your list is complete.

8. Click Save to save the changes (or Cancel to exit without changing).

Configuring Delegated Access Administrators

When the responsibility for managing the Access System falls on a few people, you may want these people to appoint others to share the work. People currently responsible for resources generally know best to whom to delegate responsibility. The ability to delegate Access System administration to other people enables you to scale administration of your resources, empowering those closest to the resources and most knowledgeable about them to manage them.

A Master Access Administrator can create a group of users and assign administrative rights to the group. The Master Access Administrator can assign the same administrative rights to more than one group. For example, Group1 and Group2 can both be assigned the right to manage Access Servers.

The following functions can be delegated:

- Add, modify, delete AccessGate configurations.
- Add, modify, delete Access Server configurations.
- Add, modify, delete Access Server clusters.
- Add, modify, delete authentication schemes.
- Add, modify, delete authorization schemes.

- Add, modify, delete host identifiers.
- Add, modify, delete resource type definitions.
- Modify the revoked user list.

  To manage the revoked user list, a delegated administrator must have access to the searchbase containing the entry for the user and must have appropriate attribute read permissions.

You can add a user to more than one group. For example, if you create one group of Delegated Administrators to manage authentication schemes and authorization schemes, and another group to manage Access Servers and Access Server clusters, the same user can belong to both groups.

When an administrator performs certain tasks, NetPoint creates an informational log. See *Volume 1* for details.

Note: Policy domain administration can also be delegated. See "Managing Server Settings" on page 26 for details.

Creating a Group of Delegated Access Administrators

The procedure below illustrates how to add Delegated Administrators to the NetPoint Access System.

To create a group of Delegated Access Administrators

1. From the Access System Console, click System Configuration > Configure Admins.

   The Configure Administrators page appears.

2. Under the title Groups of Delegated Administrators, click the Add button.

   The Create a New Group of Delegated Administrators page appears. You can complete all information requested or create an empty group with no administrative rights or members.

3. Provide the information requested.

4. For example:

   Name—A name for this group
   Description—Optional description
   Administrative Rights—Select the rights you want to give to this group

5. Click the Select User button, beside the Members label, to display the Selector.

6. Use the Selector to add people to this group, then click Done when you are finished to return to the Create a new group of Delegated Administrators page.

7. Click Save to complete the process.

Modifying a Group of Delegated Administrators

The procedure below illustrates how to alter a group of Delegated Administrators in the NetPoint Access System.

To modify a group of delegated administrators

1. From the Access System Console, click System Configuration > Configure Admins and click the link for a group.

   The Modify Group of Delegated Administrators page appears.

2. Click Modify.

   The page changes to show editable fields for group name, description, and so on.

3. Make your changes and click Save.

Managing Server Settings

The Access System Console, System Configuration function, enables you to view and alter Access Server and directory server settings as well as add a license key, and other important items. Discussions below explain:

- "Viewing Server Settings" on page 26
- "Configuring Licenses" on page 27
- "Customizing Email Addresses" on page 28
- "Configuring an SSO Logout URL" on page 29
- "Configuring the Directory Server" on page 30

Note: Only NetPoint Administrators can alter these settings.

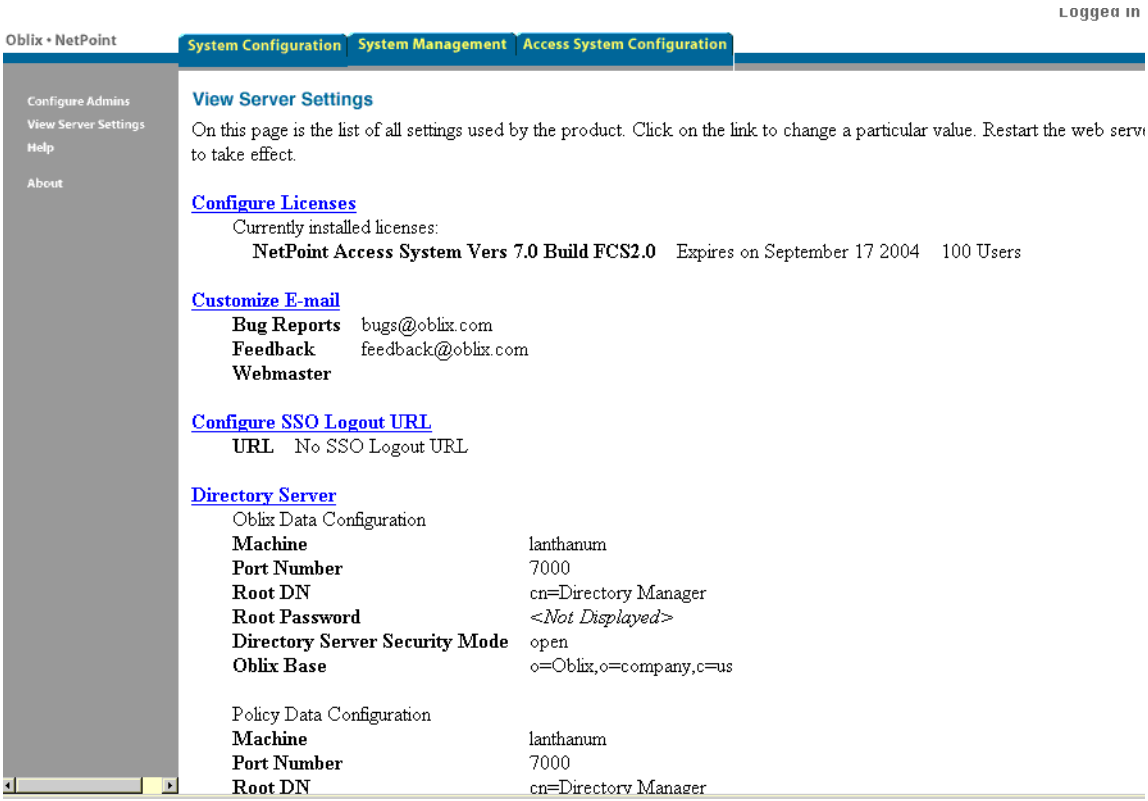
Viewing Server Settings

You use the Access System Console to view server settings, as described below.

To view server settings

1. Launch the Access System Console.
2. Click System Configuration > View Server Settings.

   The View Server Settings page appears, as shown below.

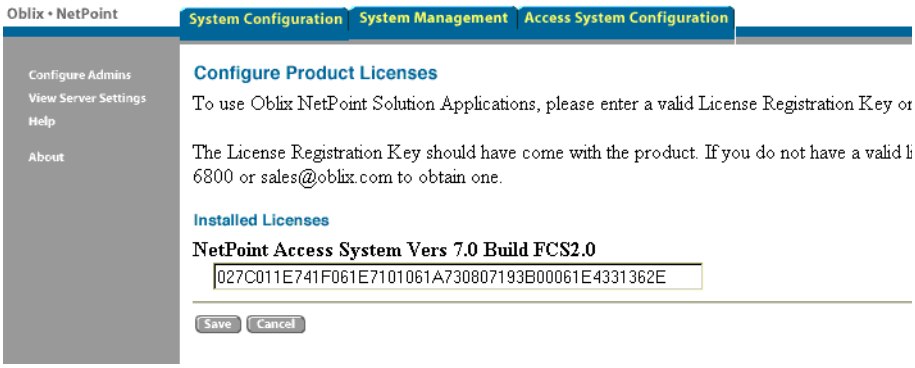


Configuring Licenses

You use the Configure Licenses function to view the licenses for the Access System, and to enter a license key when you receive the permanent key from Oblix following installation.

To configure licenses

1. Obtain a license key from Oblix.
2. Launch the Access System Console.
3. Click System Configuration > View Server Settings to display the View Server Settings page.
4. Click Configure Licenses to display this page, which displays the license keys for installed Access System instances.



In this example, the actual license key is partially blocked out.

5. Enter the new license key in the appropriate field.
6. Click Save to save your changes (or click Cancel to exit without saving).

Customizing Email Addresses

You use the Customize Email function to specify email addresses for user feedback.

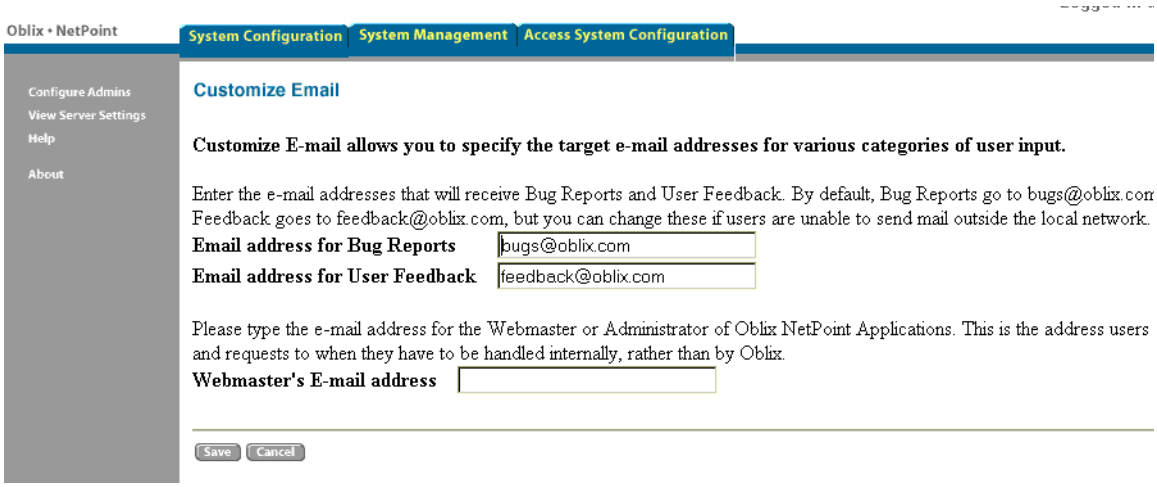
The end user accesses email addresses by clicking About on the side navigation bar, then clicking Submit Admin Feedback or Submit Oblix Feedback.

To customize email

1. In the Access System Console, click System Configuration > View Server Settings.

   The View Server Settings appears.

2. Click Customize Email to display this page.



3. Type email addresses in the following fields:

Address	Description
Email address for Bug Reports	This address must be changed to be sent to a person or alias in your organization. This person or department can either solve the problem or contact Oblix for help.
Email address for User Feedback	When a user submits an Oblix Feedback form, the data is sent to the address specified. The default is feedback@oblix.com.
Webmaster's Email address	When a user submits an Admin Feedback form, the data is sent to the address specified. The default is webmaster@company.com.

4. Click Save to save your changes (or Cancel to exit without saving).
5. Restart your Web server.

Configuring an SSO Logout URL

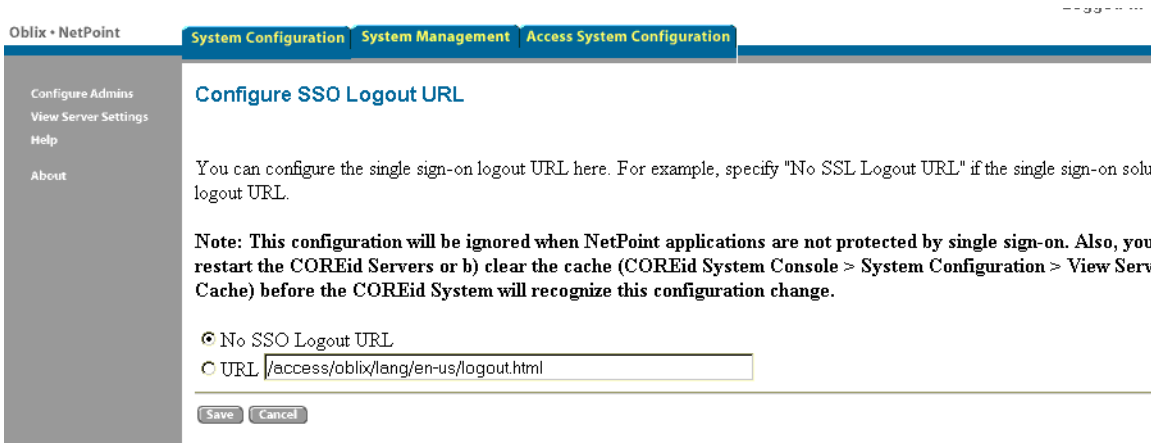
The Access System sets the ObSSOCookie for each user or application that accesses a resource protected by the Access System. The ObSSOCookie enables users to access other resources protected by the Access System that have the same or a lower authentication level. Calling the SSO Logout URL removes the ObSSOCookie, thereby requiring the user to re-authenticate the next time he or she accesses a resource protected with the Access System.

The logout.html form also contains javascript for removing the ObTemC cookie set for the COREid System. It does not, however, remove any cookies set by third-party applications. Therefore, you may need to customize this SSO logout.html to remove these cookies, to ensure users must re-authenticate. See “Configuring Single Sign-On” on page 295.

To configure the SSO Logout URL

1. In the Access System Console, click System Configuration > View Server Settings.
2. Click Configure SSO Logout URL.

   The following page appears.



3. Choose the option you want:
   - If you use a third-party program for logging users out, select No SSO Logout URL
   - If you want to have the NetPoint COREid System and Access System automatically call this page when the user clicks Logout, select URL.

   Note: You must manually create a link to this logout.html page from other resources protected by the Access System.

4. Click Save.

Note: You must flush the COREid server cache after changing the SSO Logout value. See Chapter 2, “COREid System Administration” for more information.

Configuring the Directory Server

You use the Directory Server Configuration page to modify various directory server settings using the Access System Console. This is similar to modifying Directory Server details using the COREid System Console, as discussed in *Volume 1*. Directory server details available in the Access System Console include those for Oblix data and policy data.

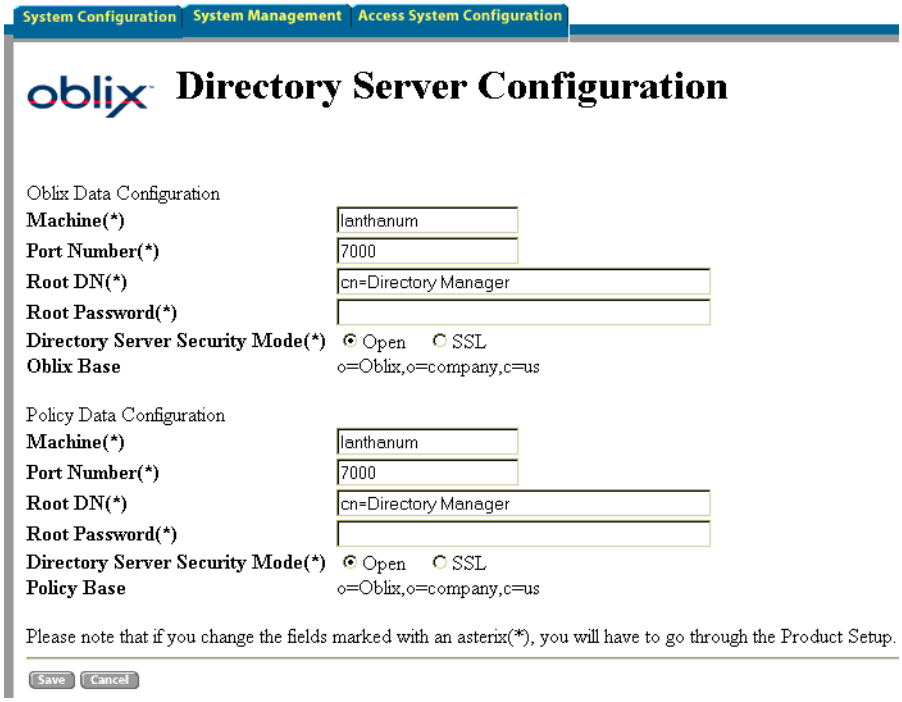
To configure the directory server

1. From the Access System Console, click System Configuration > View Server Settings.

   The View Server Settings page appears.

2. Click the Directory Server link.

   The Directory Server Configuration page appears. Notice that the page is divided into two areas: one for Oblix data configuration and one for Policy data configuration. The Oblix base and policy base on this page cannot be changed.



   The Oblix Base identifies the location of all Oblix-specific information. You cannot change this information. The Policy Base identifies the location in the DIT under which all NetPoint policy data is stored, which you cannot change.

Note: If you change the information in any field marked with an asterisk (*), you must repeat product setup as described in *Volume 1*.

---

3. Specify configuration information for Oblix configuration data, as shown in the following table.

Field	Description
Machine(*)	Name or IP address of the machine where the directory server managing the user data, configuration data, or policy data is installed
Port Number(*)	Port number of the machine on which the directory server managing the user data, configuration data, or policy data is running
Root DN(*)	Root DN of the directory server
Root Password(*)	Root password of the directory server
Directory Server Security Mode(*)	Security mode the directory server uses to protect its communications

4. Specify configuration information for Policy data, as shown in the table above.
5. Click Save to save your changes (or Cancel to exit without saving).

2 Configuring AccessGates and Access Servers

NetPoint AccessGates and Access Servers are key components when a user attempts to access a protected resource. This chapter explains how to create and configure AccessGate and Access Server instances and includes the following topics:

Prerequisites

NetPoint 7.0 should be installed and set up as described in the *NetPoint 7.0 Installation Guide*. The *Introduction to NetPoint 7.0* manual provides an overview of NetPoint not found in other manuals. Also, familiarize yourself with *Volume 1*, which provides a brief review of Access System applications and installation; introduces Access System configuration and administration; and includes common functions, configuration, and administration.

Important: You must have appropriate rights to complete activities in this chapter. See "Configuring Access Administrators" on page 22 for more information.

About Using the NetPoint Access System

Controlling access to resources—such as applications and content—is the cornerstone of e-business infrastructure. You want to allow some users to use certain resources and deny access to others.

You control access to your company's resources through the NetPoint Access System. The NetPoint Access System is controlled through a Web-based user interface that consists of the Access Manager and the Access System Console.

Access Manager—You use the Access Manager to create and manage policy domains to protect resources, and to test policy enforcement.

Access System Console—The Access System Console, a part of the Access Manager, provides functions for the following configuration and management tasks:

- **System Configuration**—Functions provide include Configure Admins and View Server Settings. The View Server Settings function provides information on licenses, email, SSO logout URL, and the directory server. For more information, see "Configuring Access Administrators and Server Settings" on page 21.
- **System Management**—Allows you to identify Access Servers on which to run diagnostics, manage user access-privilege reports, and manage sync records. For details, see "Access System Configuration and Management" on page 323.
- **Access System Configuration**—Functions include operations for the following (see also "Managing Access System Configuration Files" on page 335):
  - Access Server Clusters
  - AccessGate Configuration
  - Access Server Configuration
  - Authentication Management
  - Authorization Management
  - User Access Configuration is described in "Configuring User Access" on page 325.
  - Common Information Configuration
  - Host Identifiers
  - Configure NetPoint BEA ReadyRealm is described in the *NetPoint Integration Guide.*

As discussed in detail in the *Introduction to NetPoint 7.0* manual, and *Volume 1* of this guide, the following components are also integral to the NetPoint Access System.

Access Server—A standalone server on a Unix or Windows platform that provides authentication, authorization, and auditing services for AccessGates.

WebGate—An out-of-the-box AccessGate processes Web resource requests. It is a plug-in that intercepts HTTP requests for Web resources and forwards them to the Access Server for authentication and authorization.

AccessGate—A custom Access Client processes Web and non-Web resource requests from users or applications. It intercepts user requests and forwards them to the Access Server for authentication and authorization.

For more information about Access Servers and WebGates, see "The Access Login Process" on page 85.

Configuring Access Servers

As described in the *NetPoint 7.0 Installation Guide*, you must install at least one Access Server and it is recommended that you install at least two on different machines to ensure uninterrupted service to your users. Each Access Server must be configured to communicate with one or more AccessGate instances, and to communicate with a directory server.

Task overview: Creating an Access Server

1. Create an Access Server instance in the Access System Console, as described in "Adding an Access Server Instance" on page 38.
2. Install the Access Server, as described in the *NetPoint 7.0 Installation Guide*.
3. Configure the Access Server, as described in "Adding an Access Server Instance" on page 38.

Note: Access Servers record their activity in Greenwich Mean Time (GMT) because you could have servers operating in several time zones. Synchronizing clocks on NetPoint hosts is critical. See the *NetPoint 7.0 Installation Guide* for more information.

Using the Access System Console > Access System Configuration function, you can perform a number of key tasks including those described below:

- "Viewing Access Server Configuration Details" on page 36
- "Adding an Access Server Instance" on page 38
- "Modifying Access Server Details" on page 43

- “Deleting an Access Server” on page 44
- “Clustering Access Servers” on page 44
- “Managing Access Servers from the Command Line” on page 47

Important: You must have appropriate rights to configure the Access System. See “Configuring Access Administrators” on page 22 for more information.

Viewing Access Server Configuration Details

You can view all the configured Access Servers in the Access System Console.

To view Access Server configuration details

1. Launch the Access System Console.
2. Click Access System Configuration > Access Server Configuration.

   The existing Access Servers are listed on the page.
3. Select an Access Server to view its configuration.

   The configuration details of the Access Server appear, as described next. For details about configuring these parameters, see “Adding an Access Server Instance” on page 38.

Access Server Configuration Parameters

The Access Server configuration parameters available through the Access System Console, Access Server Configuration function, are as follows:

Field	Description
Name	Name of the Access Server.
Hostname	Name of the Web server that is hosting the Access Server.
Port	Port number the Access Server is listening to.
Debug	Indicates whether debugging is on or off.
Debug File Name	The name of this Access Server’s debug file. The absolute path to the debug file is also indicated. If the file does not exist, it is created after you restart the Access Server.

Field	Description
Transport Security	Level of transport security to and from the Access Server. Available options are: **Open**—No transport security. **Simple**—Encrypted transport security with prepackaged certificates. **Cert**—Encrypted transport security.
Maximum Client Session Time	The duration, in hours, for a connection between AccessGate and an Access Server.
Number of Threads	Maximum number of threads allowed on the Access Server.
Access Management Service	Whether Access Management service is enabled. Setting this to On enables the Access Management engine in the server. The Access Server starts servicing Access Manager API requests from AccessGates for policy management.
Audit File Name	Path to this Access Server's audit file.
Audit File Size	Maximum size of the audit file, in bytes.
Buffer Size	Size of the audit buffer, in bytes.
File Rotation Interval	Time, in seconds, that this audit file can exist.
Engine Config Refresh Period	Frequency, in seconds, of configuration updates to this server. **Note:** Changes you make to this parameter do not take effect until the previous Engine Config Refresh Period has expired.
URL Prefix Reload Period	Frequency, in seconds, with which new URLs are recognized by this Access Server.
Password Policy Reload Period	Frequency, in seconds, with which new password policies are recognized by this Access Server.
Maximum Elements in User Cache	Number of authenticated users that can be saved in the Access Server's cache.
User Cache Timeout	Maximum time, in seconds, for inactive user data to reside in cache.

Field	Description
Maximum Elements in Policy Cache	Maximum number of elements that can be stored in the policy cache.
Policy Cache Timeout	Maximum time, in seconds, for inactive policy data to reside in cache.
SNMP State	Specifies whether SNMP is enabled or not. For details on SNMP monitoring, see *Volume 1*.
SNMP Agent Registration Port	The port number of the SNMP agent.

Adding an Access Server Instance

Before installing the Access Server, you must add a new instance in the Access System Console. At this time, you need only specify the Access Server name, hostname, port, and transport security mode. After installation, you can configure the instance further.

Note: You must add the Access Server instance to the Access System Console before installing the component.

The procedure below describes how to add and completely configure the instance.

To add an Access Server instance

1. From the Access System Console, click Access System Configuration > Access Server Configuration.

   The Access Server Configuration page appears.

2. Click Add.

   The Add a New Access Server page appears.

3. This image shows the top portion of the page.

You must enter information in the Name, Hostname, and Port fields. All other fields are optional.

4. In the Name field, type a name for this server.

   Type an alphanumeric string without spaces.

   Note: You cannot give the same name to an AccessGate and an Access Server.

5. In the Hostname field, type the name or IP address of the computer hosting this server.
6. In the Port field, type the port number of the computer hosting this server.
7. In the Debug field:
   - Click On to capture all messages sent from each AccessGate and Access Manager to this Access Server.

     The messages are stored in a Debug file if a Debug file is provided. Otherwise, the messages are printed out to *stderr*.
   - Click Off if you do not want to capture this information.

     Capturing messages confirms that communication is taking place between AccessGate instances and this Access Server.

   Important: Capturing debug messages records user passwords, a potential security problem, and causes the Access Server log file to grow rapidly. Use debugging only when diagnosing a problem.

8. In the Debug File Name field, type the path to this Access Server's debug file.

9. In the Transport Security field, select a method for encrypting messages between this Access Server and the AccessGates it is configured to talk to.

   For AccessGates and Access Servers that are configured to communicate with each other, be sure to choose the same encryption method.

   Your choices are Open mode, Simple mode, or Cert mode.

   For a description of configuring transport security modes, see *Volume 1*.

10. In the Maximum Client Session Time (hours) field, type the number of hours that a connection between an AccessGate and this Access Server can last.

    The default is 24 hours.

    The longer the session time, the more vulnerable your system is to attack.

11. In the Number of Threads field, type the number of threads this Access Server will start.

    The default entry is 100. The minimum is 1.

12. Beside Access Management Service, select the option for your environment.

    Setting the Access Management Service to "On" enables the Access Management engine in the server. The Access Server starts servicing Access Manager API requests from AccessGates for policy management.

13. Choose the Auditing option you want for your environment:

    - **Audit to Database** (on/off)— Selecting "On" requires specific configuration in NetPoint plus installation of a supported database, as described in *Volume 1*.
    - **Audit to File** (on/off)—Selecting "On" requires specification of the following additional items:
      - *Audit File Name*—Type the path to this Access Server's audit file. The file is stored on the computer hosting this Access Server. Each Access Server has its own audit file. The information captured for the file is determined by the Master Audit Rule. See "Configuring the Master Audit Rule" on page 132 for information details.
      - *Audit File Size (bytes)*—Type a number representing the number of bytes this audit file can hold. If you change the default size, you need to restart the server after committing the change. When the maximum size is reached, the current file is closed and stamped with the date and time, and a new audit file with the original name is created.

14. In the Buffer Size (bytes) field, type a number representing the number of bytes that the audit file's buffer can hold.

    The default is 512000. Using a buffer increases performance by reducing the number of times information is read to the audit file.

- If you type a value in this field, audit information is stored in the buffer before it is written to the audit file. When the buffer reaches the size you specify, it transfers its contents into the audit file.
- If you enter 0 in this field, audit information is written directly to the audit file.

15. In the File Rotation Interval (seconds) field, type a number representing the number of seconds that this audit file can exist.

    The default is 0. A setting of 0 means that the audit file never times out, and audit information continues to be added to the file.

    When the limit is reached, this file is "rotated," which means it is closed, stamped with the date and time, and a new audit file with the original name is created.

16. In the Engine Config Refresh Period (seconds) field, type a value in seconds to indicate the frequency with which configuration changes to this server take place.

    The default is 14400. A setting of 14400 means the audit file name and related parameters are refreshed once every 4 hours. If set to 0, they are never refreshed. They are loaded when the server comes up and remain the same while the server is up.

    The changes are implemented with the frequency you indicate in this field. For example, if you type 600 seconds, configuration changes are implemented within 10 minutes.

    For more information, see "Modifying Access Server Details" on page 43.

    Note: Changes you make to this parameter do not take effect until the previous Engine Config Refresh Period has expired.

17. In the URL Prefix Reload Period (seconds) field, type a number representing the frequency with which new URLs are recognized by this Access Server.

    The default is 7200.

    For example, if you type 600 in this field (600 seconds = 10 minutes), URLs are reloaded from the directory server every 10 minutes. This is helpful in cases where a particular URL Prefix cache flush request did not reach an Access Server.

18. In the Password Policy Reload Period (seconds) field, type a number representing the number of seconds that specifies the reload interval for the password policies.

    The default is 7200.

19. In the Maximum Elements in User Cache field, type a number representing the number of authenticated users that can be saved in the Access Server's cache.

    The default is 10000.

    When the maximum is reached, the newest user activity is added to the cache, and the oldest is deleted.

20. In the User Cache Timeout (seconds) field, type a number representing the number of seconds that entries remain in the user cache until they are purged.

    The default is 1800 (30 minutes). Setting the timeout to 0 means that the cache element never expires.

    After the timeout on cached user entry has expired, the Access Server goes to the directory server to get user profile data needed during authentication actions and authorization actions.

21. In the Maximum Elements in Policy Cache field, type a number representing the number of items and activities associated with activities within policy domains, such as URLs mapped to specific rules, that can be cached.

    The default is 100000.

    When the maximum is reached, the newest user activity is added to the cache, and the oldest is deleted.

22. In the Policy Cache Timeout field, type a number representing the number of seconds that entries about policies can last before they are purged.

    The default is 7200 (two hours).

23. In the SNMP State field, click On to enable SNMP or click Off to disable SNMP.

24. In the SNMP Agent Registration Port field, enter the port number for the SNMP Agent.

25. Click Save to save this new Access Server (or click Cancel to exit the page without saving).

26. Repeat the above steps for each Access Server you want to install in your e-business infrastructure.

Now that you have created an Access Server instance, you can install this Access Server. When installing, use the Name, Hostname, and Port number information you typed in this page.

See the *Netpoint 7.0 Installation Guide* for information on installing an Access Server.

Configuring a Directory Server Profile for the Access Server

A default directory profile is created for the Access Server during Access Server installation. For information on how to view or modify this profile, see information on directory profiles in *Volume 1* of this guide.

If you install more than one Access Server instance, each server uses the same default directory server profile. If you modify a shared directory server profile for a particular Access Server instance, all of the other Access Server instances are affected. If you do not also change the profiles for these servers, you receive a warning whenever you:

- View the server configuration
- Restart the server
- Reconfigure the server

Modifying Access Server Details

Occasionally, you may need to change an Access Server's configuration settings. You can modify an Access Server instance through the Access System Console. If you change any field marked with an asterisk, you must restart the Access Server.

Note: You cannot change an Access Server's name. To give an Access Server instance a new name, you must delete and uninstall the current instance, then create a new one.

To modify an Access Server

1. Launch the Access System Console.
2. Navigate to Access System Configuration > Access Server Configuration.

   The Access Server Configuration page appears. The name, host, and port of each configured Access Server are listed on this page.
3. Click the link of the Access Server you want to modify.
4. On the Details for Access Server page, click Modify.

   The Modify Access Server page appears. For details about all parameters, see "Access Server Configuration Parameters" on page 36.
5. Enter new values (see "Adding an Access Server Instance" on page 38.)
6. Select Update Cache to immediately send your changes to this Access Server's cache.
7. Click Save to save your changes (or click Cancel) and return to the previous page.

Deleting an Access Server

To remove an Access Server from your system, you must first delete its configured instance, then uninstall the Access Server.

When you delete an Access Server, all AccessGate instances that are configured to send requests to the server are automatically notified. Before deleting an Access Server, make sure that all AccessGate instances are configured to send requests to at least one other Access Server.

To delete an Access Server

1. Launch the Access System Console.
2. Click Access System Configuration > Access Server Configuration.

   The existing Access Servers are listed on the page.
3. Select the server you want to delete.
4. Click Delete.

   You are prompted to confirm your decision.
5. Click OK to delete the instance (or click Cancel to stop the deletion).
6. Uninstall the Access Server.

Clustering Access Servers

In large NetPoint implementations, there can be thousands of AccessGates. Whenever a new Access Server is added, the administrator must manually configure all the AccessGates to communicate with the Access Server. In addition, the administrator must also configure failover and load balancing for the new Access Server, as described in the *NetPoint 7.0 Deployment Guide*.

Grouping Access Servers into *clusters* reduces the time needed to manage these tasks, because NetPoint automatically performs some of the configuration tasks. After you create a cluster, you add Access Servers to it and then associate one or more AccessGates with the cluster. NetPoint automatically configures all the AccessGates associated with the cluster to communicate with all the Access Servers in the cluster.

Managing Access Server Clusters

If you are a Master Access Administrator or a Delegated Access Administrator with appropriate rights to manage Access Server clusters, you can:

- Add an Access Server to multiple clusters.
- Associate multiple AccessGates with a cluster
- Associate multiple clusters with an AccessGate.

NetPoint dynamically configures failover and load balancing for all the servers in a cluster and ensures that requests are routed to those Access Servers with the lightest load. For details about configuring failover, see the *NetPoint 7.0 Deployment Guide*.

Note: All Access Servers in a cluster and all AccessGates associated with the cluster *must* have the same transport security mode and Access Manager Service state.

To add an Access Server cluster

1. Launch the Access System Console.
2. In the Access System Configuration page, click Access Server Clusters.

   The existing Access Server clusters are listed on the page.
3. Click Add.

   The Create a New Cluster of Access Servers page appears.
4. Enter a unique name for the cluster.
5. Select a transport security mode for the cluster.

   Open mode is the default. You can select Open, Simple, or Cert. All Access Servers in a cluster must have the same transport security mode.
6. Specify the Access Management Service State.
   - **On**—Click the On button.
   - **Off**—By default, it is turned Off.

   Note: All Access Servers in a cluster must have the same Access Management Service State.

   ---
7. Click Next to go to the next page (or click Cancel if you do not want to save the cluster).

8. On the next page, select the Access Server that you want to add to the cluster by clicking an Access Server in the list.

   The Access System only displays those servers that have the same security mode and AM Service State as the one specified for the cluster.

9. Click the >> button to add the Access Server to the cluster (to remove an Access Server from the cluster, select it in the Access Servers in Clusters box and click the << button).

10. Click Save to save your changes (or click Cancel if you do not want to save your changes); click Back to return to the first page.

Note: If you click Back and change the transport security mode or the AM Service State, then click Next, you have to re-select Access Servers with the new security mode or AM Service State.

To view or modify an Access Server cluster

1. Launch the Access System Console and click Access Server Clusters.

   The existing Access Server clusters are listed on the page.

2. Click a cluster to view its details.

   The Details for NetPoint Access Server Cluster page appears. The details of Access Servers in the cluster are listed.

3. Click Modify to modify a cluster's details.

   The Modify Cluster page appears.

4. You can add or delete Access Servers.

   - To add an Access Server to the cluster, select the server from the Available Access Servers list and click the >> button to add it to the cluster.
   - To remove an Access Server from the cluster, select the server in the Access Servers in Cluster box and click the << button to remove it from the cluster.

5. Click Save (or click Cancel if you do not want to save your changes).

Managing Access Servers from the Command Line

You can perform an automated installation of the Access Server using a file that contains installation parameters and values. This is called installing in *silent mode*. Silent mode permits installation without user intervention.

To install an Access Server in silent mode

1. At the command line, enter the following command:

   -S -f *aaa_input.lst* file

   where *aaa_input.lst* is a file that contains installation parameters and values.

NetPoint provides a sample input file named silent-mode-sample-AAA-Input.lst. The file is located in:

AccessServer_install_dir\access\oblix\tools\configureAAAServer

where *AccessServer_install_dir* is the directory in which the Access Server is installed.

2. For more information on Silent mode, see the *NetPoint 7.0 Installation Guide*.

Using the ConfigureAAAServer Tool

You can perform Access Server-related administration tasks through a command-line tool called configureAAAServer. This tool can be used in both Windows and Solaris installations.

Commands that you can use with configureAAAserver tool:

- install
- reconfig
- chpasswd
- remove

Windows Systems—Use the remove and install commands to remove or re-install an Access Server Service.

Non-Windows Systems—Use the **start_configureAAAServer** script to invoke the configureAAAServer tool. To see the options, you can run this tool without any options.

To access the configureAAAserver tool

1. Navigate to the folder where configureAAAserver is located.

   The default location is:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

   Note: On non-Windows systems, use start_configureAAAServer.

2. Use the configureAAAserver tool in a procedure, as needed.

To re-configure an Access Server

1. Navigate to the folder where configureAAAserver is located.

   For example:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

2. Run the following executable:
   `configureAAAServer reconfig` *`AccessServer_install_dir`*
3. Select Yes (Y).
4. Specify the following when prompted:
   - The transport security mode in which you want Access Server to run
   - The transport security mode in which the directory server is running
   - The host machine on which the directory server resides
   - The port number on which the directory server listens
   - The Bind DN of the directory server
   - The password of the directory server
   - The directory server to which you are connecting
   - The location where Oblix data is stored
   - The Configuration DN
   - The Policy Base
   - The Access Server ID

   See “Configuring the Directory Server” on page 30 for information on directory server configurations.
5. Restart the Access Server.

To modify common parameters

1. Navigate to the folder where configureAAAserver is located.

   The default location is:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

   where *AccessServer_install_dir* is the directory where the Access Server was installed.

2. Run the following executable:

   `configureAAAServer reconfig` *`AccessServer_install_dir`*

3. Select No (N) when you are asked if you want to reconfigure the Access Server.

   You are then be asked if you want to specify failover information for Oblix or Policy.

4. Select Yes (Y).

5. Specify whether the data is stored in the Oblix tree, or the Policy tree.

   The following options appear:

   a) Add a failover server

   b) Modify a failover server

   c) Delete a failover server

   d) Modify common parameters

   e) Quit

6. Select Modify common parameters

7. Specify values for the following common configuration parameters as needed:

   a) **Maximum Connections**—The maximum number of connections that an Access Server can establish with the associated directory servers for load balancing.

   b) **Sleep For (seconds)**—The frequency with which the Access Server checks its connections to the directory server. For example, if you set a value of 60 seconds, the Access Server checks its connections every 60 seconds from the time it comes up.

   c) **Failover Threshold**—The number representing the point when the Access Server opens a new connection to a directory server. For example, if you type 3 in this field, and the number of connections from the Access Server to the directory server falls to 2, a new connection is opened between the Access Server and the configured directory servers.

d) **Maximum Session Time**—The maximum period of time that a session between an Access Server and a directory server is valid.

For more information, see the *NetPoint 7.0 Deployment Guide*.

8. Select Quit to exit.

   You are prompted to commit the changes.

9. Select Y to commit your changes (or select N to cancel your changes).

To remove an Access Server service

1. Navigate to the folder where configureAAAserver is located.

   The default location is:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

   Note: On non-Windows systems, use start_configureAAAServer.

2. From the command line, run the following executable:

   `configureAAAServer remove` *`AccessServer_install_dir`* `serviceName`

   where *AccessServer_install_dir* is the directory in which the Access Server was installed and *serviceName* is the name of a service such as NetPoint_AccessServer.

   A message appears stating that the registry entries are being removed. This confirms that the Access Server has been removed.

   Note: The *serviceName* variable is applicable only for MS Windows. The *serviceName* is the name you specify for the Access Server on the Access System Console.

To re-install an Access Server service

1. Navigate to the folder where configureAAAserver is located.

   For example:

   AccessServer_install_dir\access\oblix\tools\configureAAAServer

   Note: On non-Windows systems, use start_configureAAAServer.

2. From the command line, run the following executable:

   `configureAAAServer install` *`AccessServer_install_dir`* `serviceName`

   where *AccessServer_install_dir* is the directory in which the Access Server was installed and *serviceName* is the name of a service such as NetPoint_AccessServer.

3. Specify the following:

- Do you want to reconfigure the Access Server?
- The transport security mode for the Access Server
- The transport security mode for the Oblix directory server
- The host machine on which the Oblix directory server resides
- The port number on which the Oblix directory server resides
- The bind DN of the Oblix directory server
- The password of the Oblix directory server
- The Oblix directory server to which you are connecting
- The configuration DN
- The location of the policy data
- The policy base
- The Access Server ID

4. Note the name of the Access Server service.

   A message appears stating that the Access Server has been successfully installed.

5. Start the Access Server from the Windows Control Panel services.

Setting Number of Queues from the Command Line

Requests are queued as they are sent to an Access Server. A *thread* processes each request. For example, if you have two request queues and 60 threads, the Access Server spawns 120 threads.

You cannot specify the number of queues in the Access System Console. When you configure an Access Server, however, you specify the number of threads in the Number of Threads field. The default setting is 60.

Use a command line entry to specify the number of queues each Access Server can support. Keep the number of queues in balance with the number of threads. Typically, one queue per WebGate is adequate.

A command is available with Solaris, Windows 2000, or Windows NT. On Solaris, you open a shell window to use this command. On Windows, use the Start Parameter field in the Services window to use this command.

To set the number of queues on Solaris

1. Open a shell window.
2. At the command line, enter the following command:

```
start_access_server -QN
```

where *N* is the number of queues.

To set the number of queues on Windows 2000

1. Navigate to Start > Programs > Administrative Tools > Services > NetPointAAAServer*ID*

   where *ID* is the name of the Access Server.

2. Right-click NetPointAAAServerID and select Properties.

   The Properties window appears.

3. To specify the number of queues, in the General tab, enter

   -Q*N*

   where *N* is the number of queues in the Start Parameter field.

To set the number of queues on Windows NT

1. Navigate to Start > Control Panel > Services > NetPointAAAServer*ID*

   where *ID* is the name of the Access Server.

2. Right-click NetPointAAAServerID and select Properties.

   The Properties window appears.

3. To specify the number of queues, in the General tab, enter

   -Q*N*

   where *N* is the number of queues in the Start Parameter field.

Configuring AccessGates

At least one Access Server and one WebGate/AccessGate must be configured and installed for the Access System to run. The Access Server must be installed before you install the AccessGate.

Task overview: Create an AccessGate

1. Create one or more AccessGate instances in the Access System Console, as described in “Adding an AccessGate” on page 60.
2. Associate each AccessGate instance with an Access Server, as described in “Associating AccessGates with Access Servers” on page 74.
3. Install an AccessGate for each instance you created in the Access System Console, as described in the *NetPoint 7.0 Installation Guide*.
4. See “Modifying a WebGate” on page 70 for additional details.

Viewing AccessGates

You can view existing AccessGates in the Access System Console by searching for the AccessGate.

You use a Search page to search for an AccessGate by any of its attributes. Depending on the attribute you select, the search conditions and values vary. For example, if you select Security as your search attribute, “Equals” may be the condition that is displayed in the drop-down list, and the possible values would be one of the available security modes. The system then displays the existing AccessGates that have been configured with the specified security mode.

The search attributes, conditions, and values for an AccessGate are listed in Table 1 below.

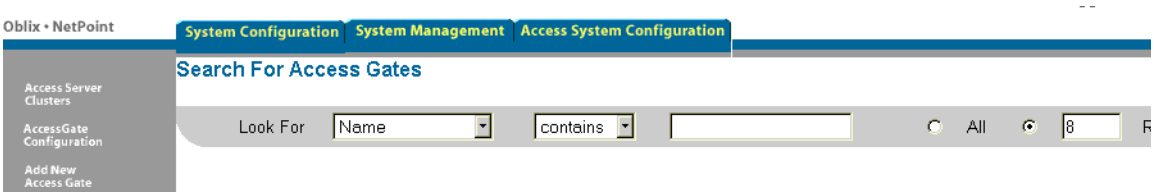
Table 1 AccessGate Search Attributes, Conditions, and Values

Attribute	Condition	Input Type	Description
Name	• Contains • Starts with • Ends with • Equals	Text box	Searches by AccessGate name.
Host Name	• Contains • Starts with • Ends with • Equals	Text box	Searches for AccessGates that are installed on the specified host machine.
Description	• Contains • Starts with • Ends with • Equals	Text box	Searches for AccessGates with a description field that contains a matching string.
Security Mode	Equals	Radio Button: Open, Simple, and Cert	Searches for AccessGates based on the transport security mode configured for them.
AM Service State	Equals	Radio Button: On or Off	Searches for AccessGates based on whether the Access Manager service has been started (On) or stopped (Off).
State	Equals	Radio Button: Enabled and Disabled	Searches for AccessGates that are enabled or disabled.

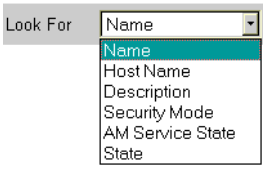
To view AccessGates

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.



   The Look For drop-down list contains a selection of attributes that can be searched, as described in Table 1 on page 53.



   The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

2. Select the search attribute and condition from the drop-down lists (or click the All button to find all AccessGates).

3. Click Go.

   The search results are displayed on the page.

4. Click an AccessGate's name to view its details.

   The configuration details of the AccessGate appear.

To view the AccessGates associated with an Access Server

1. Launch the Access System Console and click Access System Configuration > Access Server Configuration.

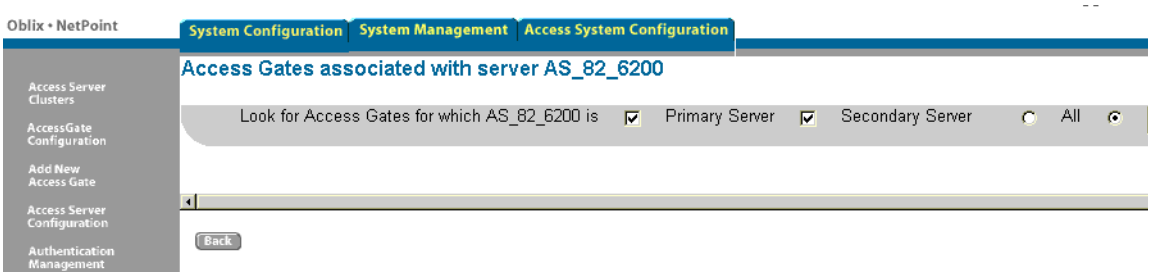
   The Access Server Configuration: List all Access Servers page appears.

2. Click the link for the desired Access Server.

   The Details for Access Server page appears.

3. Click the View Associated AccessGates button at the bottom of the Details for Access Server page.

The AccessGates associated with server page appears.



4. Check Primary Server to view AccessGates for which the Access Server is configured as a primary server.
5. Check Secondary Server to view AccessGates for which the Access Server is configured as a secondary server.
6. Select All to list all the specified AccessGates, or enter a number to specify the number of search results you want displayed on the page.
7. Click Go to display the search results.

   The details of the AccessGates associated with the Access Server are displayed on the page.
8. If there are multiple pages, click Next to go to the next page, or click Previous to go back to the previous page.
9. Click Back to return to the Access Server page.

AccessGate Configuration Parameters

AccessGate configuration parameters are shown next:

Field	Description
AccessGate Name	Name of the AccessGate.
State	Whether or the AccessGate is enabled or disabled.
Hostname	Name of the machine hosting the AccessGate.
Port	Identifies the Web server port protected by the AccessGate when deployed as a WebGate. This field should be empty when the AccessGate describes the configuration in support of an application using the Access Server SDK or Ready Realm for BEA.
AccessGate Password	Password for the AccessGate. When the AccessGate connects to an Access Server, it uses the password to authenticate itself to the Access Server. This prevents unauthorized AccessGates from connecting to Access Servers and obtaining policy information.
Debug	Turns debugging on or off.
Access Management Service	Determines whether Access Management service is enabled. • Should be enabled only if the AccessGate is using the Access Management API. The Access Management API is used for features such as COREid to Access Server Cache Flush and Self Registration Auto Login features. If these features are not being used, then the Access Management Service should not be enabled for AccessGates. • Should *not* be enabled if the AccessGate is using only the Access Server API (for example, ObResourceRequest, ObUserSession, ObAuthenticationScheme, ObConfig methods). Without the Access Management Service, can communicate with Access Servers that do have the service enabled. Refer to the *NetPoint 7.0 Developers Guide* for a discussion of the ObResourceRequest, ObUserSession, ObAuthenticationScheme, ObConfig methods. Select On to enable the server's Access Management engine. The Access Server then starts servicing Access Management API requests from AccessGates for policy management.

Field	Description
Maximum User Session Time (seconds)	Maximum amount of time in seconds that a user's authentication session is valid, regardless of their activity. At the expiration of this session time, the user is re-challenged for authentication. This is a forced logout. Default = 3600 A value of 0 disables this timeout setting.
Idle Session Time (seconds)	Amount of time in seconds that a user's authentication session remains valid without accessing any AccessGate protected resources. Default = 3600 A value of 0 disables this timeout setting.
Primary HTTP Cookie Domain	Web server domain on which the AccessGate is deployed, for instance, .mycompany.com. The Access System uses this parameter to create the ObSSOCookie NetPoint Authentication cookie. This parameter defines which Web servers participate within the cookie domain and have the ability to receive and update the ObSSOCookie. The WebGate cookie domain parameter is not used to populate the ObSSOCookie; rather it defines which domain the ObSSOCookie is valid for, and which Web servers have the ability to accept and change the ObSSOCookie contents.
Preferred HTTP Host	Defines how the hostname appears in all HTTP requests as they attempt to access the protected Web server. The hostname within the HTTP request is translated into the value entered into this field regardless of the way it was defined in a user's HTTP request. See also, "Using Preferred Hosts or Host Identifiers" on page 79.
Maximum Connections	Maximum number of connections this AccessGate can establish. This parameter is based on how many Access Server connections are defined to each individual Access Server. This number may be greater than the number allocated at any given time. Default = 1
Transport Security	Level of transport security to and from the Access Server, can be set to: • **Open**—No transport security • **Simple**—SSL v3/TLS v1.0 secure transport using dynamically generated session keys • **Cert**—SSL v3/TLS v1.0 secure transport using server side x.509 certificates
Maximum Client Session Time (hours)	Connection maintained to the Access Server by the AccessGate.

Field	Description
Failover Threshold	Number representing the point when this AccessGate opens connections to Secondary Access Servers. If you type **30** in this field, and the number of connections to primary Access Servers falls to 29, this AccessGate opens connections to secondary Access Servers.
Access Server Timeout Threshold	Number in seconds to wait for a response from the Access Server. If this parameter is set, it is used as an application TCP/IP timeout instead of the default TCP/IP timeout. **Note**: This parameter replaces the WaitForFailover parameter. The WaitForFailover parameter is used only for backward compatibility with NetPoint 5.x.
Sleep For (seconds)	Number in seconds that represents how often this AccessGate checks its connections to Access Servers. For example, if you set a value of 60 seconds for the Sleep For parameter, AccessGate checks its connections every 60 seconds from the time it comes up.
Maximum Elements in Cache	Number of cache elements NetPoint maintains. Cache elements would be URLs and Authentication Schemes. Default = 10000
Cache Timeout (seconds)*	Amount of time cached information remains in the AccessGate cache when neither used nor referrenced. Default = 1800
Impersonation Username	The name of the trusted user that you created to be user for impersonations. You specify the trusted username here to bind it to this AccessGate (NetPoint WebGate) so that the AccessGate can use it for impersonation. For information about impersonation and explanation of how to create a trusted user for impersonation, see “Enabling Impersonation with NetPoint” on page 381.
Impersonation Password	The password for the trusted user to be used for impersonation. You must enter this password twice; that is, you are asked to re-type it.

Adding an AccessGate

You must add an AccessGate instance in the Access System Console before installing the AccessGate. Required parameters before installation include the AccessGate name, hostname, port, and transport security mode. All other AccessGate parameters can be configured after installation, as described below.

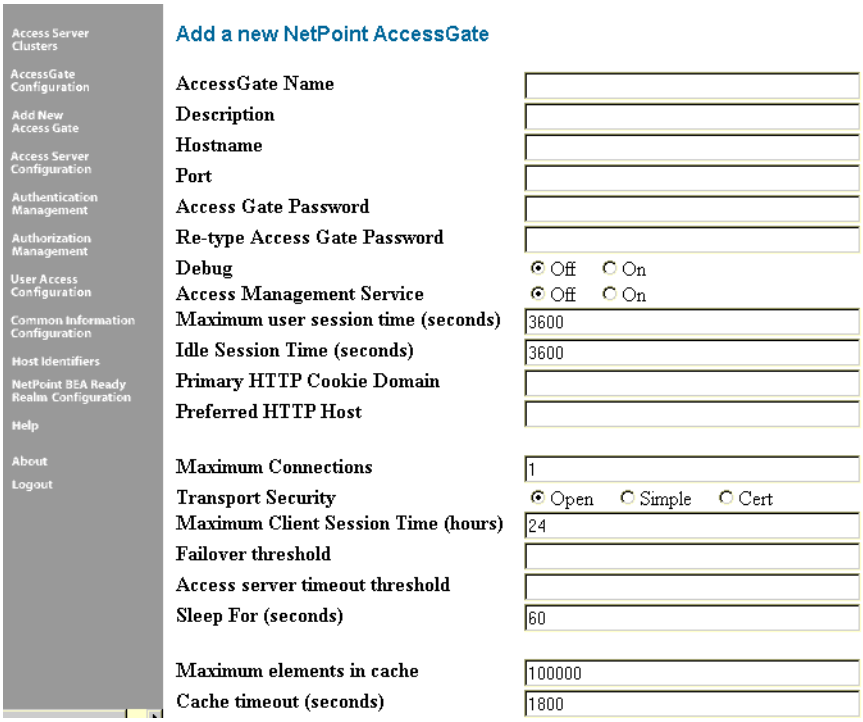
See “AccessGate Configuration Parameters” on page 57 for details about all parameters. See the *NetPoint 7.0 Installation Guide* for details about installing an AccessGate.

Important: Once you assign and save an AccessGate name, you cannot change the name. To rename an AccessGate, you must delete and uninstall the instance, then create a new AccessGate.

To create an AccessGate instance

1. Launch the Access System Console and click Access System Configuration.
2. Click Add New AccessGate.

   The Add a new NetPoint AccessGate page appears.



3. Fill in the form as follows:

   - **AccessGate Name**—Type the name of this AccessGate instance. Type an alphanumeric string without spaces. Note that an AccessGate and an Access Server cannot have the same name.
   - **Hostname**—Type the name or IP address of the server hosting this AccessGate.
   - **Port**—Type the Web server port protected by the AccessGate when deployed as a WebGate.

     This field is optional. It is recommended that you enter the Web server port number for a WebGate. Other AccessGates may or may not use a port.

     This field must be empty when the AccessGate describes the configuration in support of an application using the Access Server SDK or Ready Realm for BEA.
   - **AccessGate Password**—Type an alphanumeric string to represent a password for this AccessGate.

     The AccessGate uses this password to identity itself to an Access Server. If you provide a password here, you must enter the same password when re-configuring the AccessGate, and during AccessGate installation.

     Note: If this AccessGate is a WebGate, this password must be the same one specified when you installed the WebGate.

- **Re-type AccessGate Password**—Re-type the password.

  If the entries in these two steps do not match, when you click Save, an error message appears, and you must repeat this process.

4. The Debug field is relevant only for WebGates.

   In this field:

   - Click On to write debug messages between the AccessGate and Access Server to the standard out for most platforms. Note that on IIS this information is not written to the standard out because the IIS server runs as an NT service.
   - Click Off if you do not want to capture this information.

Important: Capturing debug messages records user passwords, a potential security problem, and causes the Access Server log file to grow rapidly. Debugging should only be turned on when diagnosing a problem.

---

5. For Access Management Service:

   - Click On to enable the AccessGate to use the Access Manager API to manage policies
   - Click Off to disable this feature.

6. Continue completing the information as follows:

   - **Maximum user session time**—Type the maximum amount of time, in seconds, that a user's authentication session is valid, regardless of their activity. At the expiration of this session time, the user is re-challenged for authentication. This is a forced logout.

     The default is 3600.

   - **Idle Session Time**—Type the amount of time in seconds that a user's authentication session remains valid without accessing any AccessGate protected resources.

     The default is 3600.

   - **Primary HTTP Cookie Domain**—Type the AccessGate's domain name.

     For example,

     ```
     .yourcompany.com
     ```

Note: The "dot" in the initial position of the domain name is required. See "Configuring Single Sign-On" on page 295 for information on how the primary HTTP cookie domain is used.

   - **Preferred HTTP Host**—Specify how the hostname appears in all HTTP requests as they attempt to access the protected Web server. The hostname

within the HTTP request is translated into the value entered into this field regardless of the way it was defined in a user's HTTP request.

You must enter one of the variations entered in the Host Identifiers feature to ensure that single sign-on works properly. For more information, see "Using Preferred Hosts or Host Identifiers" on page 79.

Note: If you are configuring a WebGate, and your browser is Internet Explorer, do not use the number 80. Using 80 as a port number can lead to operational errors in NetPoint.

- **Maximum Connections**—Type the maximum number of connections this AccessGate can establish with associated Access Servers. This number may be greater than the number allocated at any given time.

  The default is 1.

- **Transport Security**—Select a method for encrypting messages between this AccessGate and the Access Servers it is configured to talk to.

  For AccessGates and Access Servers that are configured to communicate with each other, be sure to choose the same encryption method.

  Your choices are:

  • Open

  • Simple

  • Cert

  For a description of configuring transport security modes, see *Volume 1* or see the table containing AccessGate configuration parameters on page 57.

  Note: If you want to change an AccessGate mode from simple or cert to open, you must move the /oblix/config/simple directory (if in simple mode) or the /oblix/config/*.pem files (if in cert mode) to a new folder. Then you must run the configureAccessClient program.

7. **Maximum Client Session Time**—specify the connection maintained to the Access Server by the AccessGate.

   The default is 24 hours. This value must be larger than the Sleep For parameter. Using the same session key for longer than 24 hours can make your system vulnerable to attack.

   If the Maximum Client Session Time is 0, the AccessGate establishes a new connection to the Access Server for each request that it makes to the Access Server. There may be more than one AccessGate request for each user request to the AccessGate.

   If you selected Open in the Transport Security field, this field is ignored.

8. **Failover Threshold**—Type the number representing the point when this AccessGate opens connections to secondary Access Servers. If you type 30 in

this field, and the number of connections to primary Access Servers falls to 29, this AccessGate opens connections to secondary Access Servers.

You can type a number ranging from 1 to the total number of primary servers. If you do not type a value, the number of maximum connections is used.

For details about configuring failover, see the *NetPoint 7.0 Deployment Guide*.

9. **Access Server timeout threshold**—Specify the time (in seconds) during which the AccessGate must wait for a response from the Access Server. If this parameter is set, it is used as an application TCP/IP timeout instead of the default TCP/IP timeout.

10. **Sleep For (seconds)**—Type a number (in seconds) that represents how often this AccessGate checks its connections to Access Servers.

    If a connection to an Access Server is broken, but the AccessGate finds that the Access Server is now up, it tries to reconnect to that server.

    The default is 60 seconds. The shorter the value, the quicker AccessGate can re-establish a connection to an Access Server that has come back up. But the overhead for checking connections is higher.

    An entry in this field does not affect failover to other servers, which is always immediate when needed.

11. Fill in caching details as follows:

    - **Maximum elements in cache**—Type a number of cache elements NetPoint maintains. Cache elements would be URLs and Authentication Schemes.

      The default is 10000.

    - **Cache timeout (seconds)**—Specify the time period during which cached information remains in the AccessGate cache when neither used nor referenced.

      The default is 1800.

12. Complete impersonation details, as follows:

    - **Impersonation username**—Specify the name of the trusted user that you created to be used for impersonations. You specify the user name here to bind it to this AccessGate.

    - **Impersonation password**—Type the password for the impersonation user name.

    - **Re-type impersonation password**—Type again the password for the impersonation user name.

13. Click Save to save this new instance of AccessGate (or click Cancel to return to the previous page without saving).

Now that you have created an AccessGate instance, you can install and set up this instance. When installing, use the Name, Hostname, and Port number information you typed in this page. See the *NetPoint 7.0 Installation Guide* for instructions.

Configuring Logout for a NetPoint Resource

When the COREid and Access applications are protected by a WebGate, a logout button is not automatically configured for the Access Manager and the COREid System Console. You must configure the logout button and logout URL, as explained in the following procedure.

To configure the logout button

1. From the Access System Console, click System Configuration > View Server Settings > Configure SSO Logout URL.
2. On the Configure SSO Logout URL, select the URL option and enter the logout URL.

   This URL points to a logout page that you want to show to the user when they log out of the application.

   Note: Alternatively, you can specify the logout URL on the SSOLogoutURL parameter in the OblixBaseParams.lst file. This file is located in: *AccessManager_install_dir*/access/oblix/apps/common/bin/
3. To ensure that the WebGate actually logs out the user from the COREid or Access application when they click the logout button, be sure that the LogOutUrls parameter in WebGateStatic.lst is set to the same value as the SSO Logout url.

   For example:

   WebGate_install_dir/access/oblix/apps/webgate/WebGateStatic.lst

Modifying an AccessGate

Occasionally you may need to modify an AccessGate's parameters. You can modify an AccessGate through the Access System Console or through a command line tool named *configureAccessGate*. Typically, you use the command line tool to change the transport security mode. This tool can be used in both Windows and Solaris installations.

Note: If you change fields marked with an asterisk(*), you must restart the server hosting this AccessGate. Once you assign and save an AccessGate name, you cannot change the name. To rename an AccessGate, you must delete and uninstall the instance, then create a new AccessGate.

To modify an AccessGate through the Access System Console

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.

2. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

   The search results are displayed on the page.

4. Click the name of the AccessGate you want to modify.

   The AccessGate Details page appears.

5. Click Modify.

   The Modify NetPoint AccessGate page appears. You can enter new information on this page

   You cannot change an AccessGate's name. To rename an AccessGate, you must delete it from the Access System Console and then uninstall it. You then create a new AccessGate.

6. Type new values as needed.

7. Click Save to save your changes (or click Cancel to exit the page without saving).

To modify an AccessGate through the command line

1. Navigate to *AccessGate_install_dir*\access\oblix\tools\configureAccessGate.

   where *AccessGate_install_dir* is the directory where AccessGate is installed.

2. From the configAccessGate directory, run the following command:

   `configureAccessGate -i` *`AccessGate_install_dir`* `-t AccessGate`

3. Specify parameters using the commands listed in Table 2.

Table 2 ConfigureAccessGate/ConfigureWebGate Commands

Command	Operation
[-i *install_dir*]	Specifies the installation directory for the AccessGate or WebGate.

Table 2 ConfigureAccessGate/ConfigureWebGate Commands

[-t <AccessGate|WebGate>]	Specifies whether an operation is for AccessGate or WebGate.
[-w <*AccessGate*|*WebGate ID*>]	Identifies the name of the AccessGate or WebGate.
[-m <open|simple|cert>]	Specifies the transport security mode for an operation.
[-c <request|install>]	Specifies a certificate request or installation.
[-S]	Runs configureWebGate tool in silent mode without prompting for user input. If optional parameters are not present, an error message is displayed.
[-P <*AccessGate*|*WebGate password*>]	Specifies the password for the transport security certificate for AccessGate/WebGate. This value is required only if you have specified a password for the AccessGate/ WebGate.
[-h *Access Server Host Name*]	Specifies the machine name where the Access Server installed.
[-p *Access Server Port*]	Specifies the port number of the machine where the Access Server is installed.
[-a *Access Server ID*]	Specifies the name of the Access Server.
[-r *Access Server Pass Phrase*]	Specifies the password for the Access Server. This is a global password and must be the same as the specified password for the AccessGate/WebGate.
[-Z *Access Server Retry count*]	Optional. Specifies the number of times the AccessGate/WebGate attempts to connect to the Access Server when the configureAccessGate tool is used.

To reconfigure transport security mode through the command line

1. To reconfigure an AccessGate transport security mode, run the following command:

```
configureAccessGate -i AccessGate_install_dir -t <AccessGate|WebGate> -R
```

For example:

```
configureAccessGate -i C:\NetPoint\WebComponent\access -t AccessGate -R
```

2. The system prompts you to for a transport security mode:

If you select Open. . .	If you select Simple. . .	If you select Cert. . .
The transport security mode is reconfigured to run in Open mode	1. Supply the AccessGate password. If you specified a password during installation or reconfiguration of the AccessGate, enter it. If you did not, press Enter to skip the prompt. 2. Supply the Global NetPoint Access Protocol Pass Phrase. After you enter it, the system generates and installs the certificate.	1. Supply the AccessGate password. If you specified a password during installation or reconfiguration of the AccessGate, enter it. If you did not, press Enter to skip the prompt. 2. Supply the Global NetPoint Access Protocol Pass Phrase. After you enter it, the system generates and installs the certificate. Next, complete the steps that follow.

Note: The Global Pass Phrase must always be the same for all AccessGates, WebGates, and Access Servers within an Access System installation.

For Cert mode—The system prompts you to specify whether you want to request a certificate or install a certificate.

- If you specify a certificate request, the system prompts you for the following organization information:

  Country name

  State or Province

  Locality

  Organization name

  Organizational unit

  Common Name—For example, HostName.DomainName.com

  Email address

- After you enter the above information, a certificate request is generated and placed in the *Component_install_dir*\access\oblix\config\aaa_req.pem file.

  where *Component_install_dir* is the directory in which the Access System component is installed.
- You must have this certificate request signed by the Certificate Authority.
- The system prompts you for the full paths to the location of the Certificate Key file, the Certificate file, and the Certificate Chain file.

  After you specify the paths, the transport security mode is reconfigured.

To change the transport security mode password

1. From the command line, run the following command:

   `configureAccessGate -i` *`AccessGate_install_dir`* `-t AccessGate -k`
2. Enter the following information:
   - The old password
   - The new password
   - Reconfirm the new password

The password is changed.

Deleting an AccessGate

If you delete an AccessGate, the applications and content on the hosts with which it was connected are not be protected by the Access System. Be sure this is what you want to do before deleting an AccessGate.

To delete an AccessGate

1. Uninstall the AccessGate from the host.
2. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.
3. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1 on page 53. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.
4. Click Go.

   The search results are displayed on the page.

5. Check the AccessGate that you want to delete and click Delete.

   You are prompted to confirm your decision.

6. Click OK to delete the AccessGate (or click Cancel to stop the deletion).

Managing WebGates

A WebGate is an out-of-the-box Access Client for HTTP-based resources. A WebGate is an NSAPI or ISAPI plug-in that intercepts HTTP requests for Web resources and forwards them to the Access Server.

The process of configuring a WebGate is the same as configuring an AccessGate. See “Adding an AccessGate” on page 60. Topics below provide additional information:

- “Modifying a WebGate” on page 70
- “Configuring IP Address Validation for WebGates” on page 71
- “Viewing WebGate Diagnostics” on page 71
- “Checking the Status of a WebGate” on page 73

Modifying a WebGate

Occasionally you may need to modify a WebGate’s parameters. You can modify a WebGate through the Access System Console or through a command line tool named *configureWebGate*. Typically, you use the command line tool to change the transport security mode. This tool can be used in both Windows and Solaris installations.

To modify a WebGate through the command line

1. To modify a WebGate, navigate to the directory:

   WebGate_install_dir\access\oblix\tools\configureWebGate

   where *WebGate_install_dir* is the directory in which WebGate is installed.

2. From the configureWebGate directory, run the following command:

   `configureWebGate -i` *`WebGate_install_dir`* `-t WebGate`

Specify parameters using the commands listed in Table 2 on page 66.

Example of using configureWebGate

The following is an example of configuring a WebGate using the configureWebGate tool on MS Windows:

```
C:\NetPoint\webcomponent\access\oblix\tools\configureWebGate>
configureWebGate -i c:\NetPoint\webcomponent\access -t WebGate -w andium_AG -m
cert -c install -S -P milpid -h andium -p 5160 -a andium_AS -r 99malibu -Z 5
```

Configuring IP Address Validation for WebGates

IP address validation is specific to WebGates and is used to determine whether a client's IP address is the same as the IP address stored in the ObSSOCookie generated for single sign-on.

The IPValidation parameter in the WebGateStatic.lst file determines whether or not a WebGate validates the client's IP address with the one stored in the ObSSOCookie. If IPValidation is true, the IP address stored in the ObSSOCookie must match the client's IP address; otherwise, the cookie is rejected and the user is reauthenticated.

Possible settings for this flag are true (WebGate validates the client's IP address) and false. The default is true.

Note: The IPValidationExceptions parameter lists IP addresses that are exceptions to this process. If IPValidation is true, the IP address is compared to the IPValidationExceptions list. If the address is found on the Exceptions list, it does not need to match the IP address stored in the cookie.

To change the IPValidation parameter setting to false

1. Navigate to:

   WebGate_install_dir/access/oblix/apps/webgate/WebGateStatic.lst.

   where *WebGate_install_dir* is the directory in which WebGate is installed.

2. Change the IPValidation parameter to false.

Viewing WebGate Diagnostics

NetPoint provides a WebGate Diagnostic URL to display information regarding an Access Server connected to a WebGate. It also displays associated directory server information.

Diagnostic URL links are as shown in Table 3:

Table 3 Diagnostic URL links

Domino	http://*WebGate_machine*:*portnumber*/access/oblix/apps/webgate/bin/nwebgate.cgi?progid=1
IIS	http://*WebGate_machine*:*portnumber*/access/oblix/apps/webgate/bin/webgate.dll?progid=1

Table 3 Diagnostic URL links

Netscape and Apache	http://*WebGate_machine*:*portnumber*/access/oblix/apps/webgate/bin/webgate.cgi?progid=1

where *WebGate_machine* is the machine in which the WebGate was installed and *portnumber* is the port number of the machine.

Note: For IIS6, in order to use the Diagnostic URL feature, you must enable the direct access of webgate.dll through the IIS Lockdown tool.

When you access this URL, your browser displays the following information:

Access Server		Hostname of the Access Server, its port number, and the number of connections with this WebGate
State		Status of the Access Server, either Up or Down
Created		Date and time this Access Server was installed
Install_Dir		Installation directory of this Access Server
Num Of Threads		Maximum number of threads allowed on the Access Server
Directory Information	**Directory**	Type of information stored in this directory instance, User, Policy, or Oblix
	Host:Port	Hostname and port number of this directory instance
	State	Operational status of the directory server, Up or Down
	Priority	Priority of this Access Server to this WebGate, primary or secondary
	Mode	Directory server connection mode, Open or SSL
	SizeLimit	Maximum entries the LDAP server returns for a search
	TimeLimit	How long an LDAP operation in the LDAP server runs
	LoginDN	Root DN of the directory server instance
	Created	Date and time this directory server instance was created

Checking the Status of a WebGate

Depending on the type of Web server you use, you can issue a URL to check the status of a WebGate from any browser.

To check the status of a WebGate

1. Issue one of the following URLS in the browser:

   On IIS:

   ```
   http://servername:port/access/oblix/apps/webgate/bin/
   webgate.dll?progid=1
   ```

   On iPlanet and Apache:

   ```
   http://servername:port/access/oblix/apps/webgate/bin/
   webgate.cgi?progid=1
   ```

   On Domino:

   ```
   http://servername:port/access/oblix/apps/webgate/bin/
   nwebgate.dll?progid=1
   ```

Checking the Number of Connections

If you modify the configuration of a WebGate or an AccessGate, the change takes effect in less than a minute. For example, if you add a new primary server and increase the number of connections to the Access Server by one, this happens without a re-start of the server.

The old connections between the Access Server and the WebGate (or AccessGate) are discarded after a few minutes, when any pending requests are finished. If you issue a netstat command before the old connections are discarded, you might find double the number of connections since the server was started. However, this number quickly drops to the number of configured connections, usually in a few minutes. Every time that connection information is modified, the number of connections detected on a netstat command doubles for a few minutes, then drops back to the configured number.

Associating AccessGates with Access Servers

You can associate an AccessGate with either individual Access Servers or with Access Server Clusters. For each AccessGate, you must select at least one Access Server or Access Server cluster with which it can communicate. The Access Server or the Access Server cluster must already be configured and installed.

Note: The process of associating a WebGate is the same as the process of associating an AccessGate.

You can view associated AccessGates in the Access Server details page or the Access Server Cluster details page. You can also view associated Access Servers and Access Server clusters in the AccessGate's details page. If there are any errors in the configuration between an AccessGate and an Access Server or an Access Server cluster, the error is displayed on the page.

For example, the security mode for an AccessGate could be different from the security mode of an associated Access Server or Access Server cluster. In such cases, the error is displayed on the page.

About Associating AccessGates with Clusters

When you associate an AccessGate with an Access Server cluster, NetPoint automatically configures the AccessGate to communicate with all the Access Servers in the cluster. When you disassociate an AccessGate from a cluster, the connections between the AccessGate and the Access Servers in the cluster are automatically deleted.

When you add an Access Server to a cluster, NetPoint automatically configures all the AccessGates associated with the cluster to communicate with the new Access Server. When you delete an Access Server from the cluster, the connection between the AccessGate and the Access Server is automatically deleted.

Load balancing is automatically configured, based on the number of connections that you specified when you configured the AccessGate and the number of Access Servers in the cluster. For details about configuring load balancing, see the *NetPoint 7.0 Deployment Guide*.

Failover is automatically configured when you associate an AccessGate with an Access Server cluster that you define as a primary or a backup cluster. For details about configuring failover, see the *NetPoint 7.0 Deployment Guide*.

Associating an AccessGate

Use the following procedures to associate an AccessGate with an Access Server or cluster.

Task overview: Associating an AccessGate with an Access Server or cluster includes

1. “To associate an AccessGate with an Access Server” on page 75
2. “To configure an Access Server to communicate with this AccessGate” on page 76
3. “To associate an AccessGate with an Access Server cluster” on page 76

To associate an AccessGate with an Access Server

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.

2. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1 on page 53. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

   The search results are displayed on the page.

4. Click the AccessGate of your choice.

   The AccessGate Details page appears.

   If the AccessGate is not associated with any Access Server, do the following:

   a) Click Associate Access Servers

      The Associate Access Servers with AccessGate page appears.

   b) Select Individual Servers to associate the AccessGate with an Access Server.

   c) Click Next

   The page lists all primary and secondary Access Servers configured to communicate with the AccessGate.

To configure an Access Server to communicate with this AccessGate

1. From the Access System Console, click Access System Configuration > Add New AccessGate.

   The Add a new Access Server to the AccessGate page appears.

   Note: Remember that you must configure and install an Access Server before it can receive requests from an AccessGate.

2. From the drop-down list, select an Access Server.
3. In the Select priority field, choose Primary or Secondary to specify whether the Access Server is a primary server or a secondary server.
4. Enter the maximum number of connections this AccessGate can establish to this Access Server.

   The default is 1.

5. Click Add to complete the configuration of this server, or click Back to return to the previous page.

To associate an AccessGate with an Access Server cluster

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.

2. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1 on page 53. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

   The search results are displayed on the page.

4. Click the AccessGate that you want to associate with an Access Server cluster.

   The AccessGate Details page appears.

   - If the AccessGate is not associated with any Clusters, do the following:
     - Click Associate Access Servers.

       The Associate Access Servers with AccessGate page appears.
     - Select Clusters to associate the AccessGate with clusters.
     - Click Next.

The page lists all primary and backup Access Server clusters configured to communicate with the AccessGate.

- If the AccessGate is already associated with clusters, click List Clusters

  The page lists all primary and backup clusters configured to communicate with the AccessGate.

5. Click Add to associate a Access Server cluster with the AccessGate.

   The Add a new Access Server Cluster to the AccessGate page appears.

   Note: You must configure an Access Server Cluster before it can receive requests from an AccessGate.

6. Select an Access Server cluster from the drop-down list.
7. In the Select Cluster Type field, choose Primary or Backup to specify whether the Access Server cluster is a primary cluster or a backup cluster.

   The AccessGate opens connections to the Access Servers in the primary cluster. If the AccessGate cannot open the specified number of connections, it opens connections with the Access Servers in the backup cluster.

   For details about configuring failover and load balancing, see the *NetPoint 7.0 Deployment Guide*.

8. Click Save to save your changes (or click Cancel if you do not want to save your changes).

Viewing AccessGates Associated with an Access Server

You can view AccessGates that are associated with a particular Access Server.

To view AccessGates associated with a cluster

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.

2. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1 on page 53. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

   The search results are displayed on the page.

4. Click an Access Server cluster to view its details.

   The Details for Access Server Cluster page appears.

5. In the Details for Access Server Cluster page, click View Associated AccessGates.

   The Associated AccessGates page appears.

6. Select A Primary Cluster to view AccessGates for which the cluster is configured as a primary cluster.

   Select A Backup Cluster to view AccessGates for which the cluster is configured as a secondary cluster.

7. Select All to list all the specified AccessGates or enter a number to specify the number of search results you want displayed on the page.

8. Click Go to display the search results.

   The details of the AccessGates associated with the Cluster are displayed on the page.

   If there are multiple pages, click Next to go to the next page or click Previous to go back to the previous page.

9. Click Back to return to the previous page.

Disassociating an AccessGate

Use the following procedures to disassociate an AccessGate from an Access Server or cluster.

To disassociate an AccessGate from an Access Server or an Access Server cluster

1. Launch the Access System Console and click Access System Configuration > AccessGate Configuration.

   The Search for AccessGates page appears.

2. Select the search attribute and condition from the drop-down lists, or select All to find all AccessGates.

   The Look For drop-down list is a selection list of attributes that can be searched, as described in Table 1 on page 53. The remaining fields allow you to specify search criteria that are appropriate for the selected attribute.

3. Click Go.

   The search results are displayed on the page.

4. Click the AccessGate that you want to disassociate from an Access Server or an Access Server cluster.

   The AccessGate Details page appears.

5. Choose to view server clusters or servers.

   - For the Access Server clusters associated with the AccessGate, click List Clusters.
   - For the Access Servers associated with the AccessGate, click List Access Servers.

   The Access Servers or Access Server clusters associated with the AccessGate are listed on the page.

6. Choose whether to disassociate a server cluster or a server.

   - To disassociate an Access Server cluster, select the cluster and click the Delete button.
   - To disassociate an Access Server, select the Access Server and click the Delete button.

   The connection is deleted.

Using Preferred Hosts or Host Identifiers

Web server hosts can be identified in various ways, such as a machine name or an IP address. Here are some examples of how the same host can be addressed:

- site.com
- site.com:80
- www.site.com
- www.site.com:80
- 216.200.159.58
- 111.111.11.1:80
- 3232236564 (decimal addressing)

NetPoint offers two methods for identifying Web servers that are hosting protected resources.

Note: The Preferred Host and Host Identifiers features apply only to WebGates protecting a Web server, not to all AccessGates.

A third feature, DenyOnNotProtected, can be used to deny access to all resources that are not protected by a WebGate unless access is explicitly allowed by a NetPoint rule or policy.

Important: Setting DenyOnNotProtected to true is the most secure way to protect Web server content. For more information about DenyOnNotProtected, see “Denying Access to All Resources by Default” on page 84.

See the topics below for more information:

- “Using Host Identifiers” on page 80
- “Preferred Host and Virtual Servers” on page 83
- “Denying Access to All Resources by Default” on page 84

Using Host Identifiers

As described in the previous section, a host can be known by multiple names. Use the Host Identifiers feature to enter the official name for the host, and every other name by which the host can be addressed by users. A request sent to *any* address on the list is mapped to the official host name, and applicable rules and policies are implemented.

In your Host Identifiers list:

- Each host name must be unique.
- Each host name:port number pair must be unique.
- Each host name:port number pair must belong to only one host identifier.
- Each host name:port number pair must match the end user's entry exactly.

Note: With decimal addressing it may not be practical to define all possible URL combinations for the same site.

Using the formula below to calculate possible decimal addresses for the original address 01.02.03.04, where each 0 is an 8 bit octet, you will find many ways to represent the original IP address:

Formula:

01*256^3+02*256^2+03*256+04

It may surprise you to know that the following URL values are all for the same site:

http://%61%73%74%65%72%69%78
http://%31%39%32%2E%31%36%38%2E%34%2E%32%30/
http://%33%32%33%32%32%33%36%35%36%34

For more information, you may want to look at the site http://www.karenware.com/powertools/ptlookup.asp.

Note: NetPoint does not add a default port number if the end user does not provide one.

See "To add a Host Identifier" on page 82 for the steps to create a host identifiers list.

Host identifiers can be used with virtual Web hosting. However, a disadvantage is if users type an address that is not listed, they could be allowed access to resources that you want to protect.

Following discussions include:

- "Viewing or Deleting Existing Host Identifiers" on page 82
- "Adding a Host Identifier" on page 82
- "Including Authenticating Hosts" on page 83

Viewing or Deleting Existing Host Identifiers

The Host Identifier details page displays the name, description, and hostname variations.

To view or delete existing Host Identifiers

1. Launch the Access System Console and click Access System Configuration.

   The Access System Configuration page appears.

2. In the side navigation bar, click Host Identifiers.

   The List all Host Identifiers page appears. The existing host identifiers are listed on the page.

3. To view a host identifier, select its name on the list.

4. To delete a host identifier, select its name on the list and click Delete.

Adding a Host Identifier

If you attempt to add a hostname variation that already exists for a different host identifier, a message alerting you of the duplication is displayed. You can choose to save or cancel your changes.

To add a Host Identifier

1. Launch the Access System Console and click the Access System Configuration tab.

   The Access System Configuration page appears.

2. In the side navigation bar, click Host Identifiers.

   The List all Host Identifiers page appears.

3. Click Add to add a new host identifier.

4. In the Name field, type the name of the host.

5. In the Description field, type a short description.

   Completing this field is optional.

6. In the Hostname variations field, type all possible variations for identifying this host.

   Note: When you configure an AccessGate, you must enter one of these variations in the Preferred HTTP Host field to ensure that single sign-on works properly.

7. Click Save.

Including Authenticating Hosts

If you redirect an authentication challenge to be processed by another host, you must add the name of that host to the Hostname Identifiers list. The hostname that you enter in the Challenge Redirect field must be available in the Hostname Identifiers list when adding or modifying an authentication scheme. For example, if a user is redirected to an SSL-enabled server for authentication, that server must be included.

When adding URL prefixes to a policy domain, the Delegated Access Administrator must specify a server hosting the URL prefix. When a user attempts to access a URL that is protected by the policy domain, the user is redirected to the server specified in the Challenge Redirect field for authentication.

Preferred Host and Virtual Servers

Use the Preferred Host feature to specify a hostname to which all possible methods by which the host can be addressed are redirected to the Access Server for policy evaluation. The Preferred Host feature prevents security holes that can be inadvertently created if a host's identifier is not included in the Host Identifiers list.

Note: The browser does *not* re-direct to the preferred host.

You enter the Preferred Host name in the Preferred HTTP Host field when you configure an AccessGate. For more information, see "Adding an AccessGate" on page 60. The name you enter in this field must be one of the names entered in the Host Identifiers feature. For more information, see "Using Host Identifiers" on page 80.

Preferred Host Advantage—An advantage of using a Preferred Host instead of Host Identifiers is that you do not need to enter every possible name by which a host can be addressed.

Preferred Host Disadvantage—A disadvantage is that Preferred Host cannot be used with virtual Web hosting.

The virtual Web hosting feature of many Web servers allows you to support multiple domain names and/or IP addresses that each resolve to their unique subdirectories on a single virtual server.

For example, you can host abc.com and def.com on the same virtual server, each with its own domain name and unique site content. You can have name-based or IP-based virtual hosting.

When a client makes a connection, the IP address to which the client connects is looked up in the internal IP hash table. If the lookup is successful, then the doc root of that IP is served.

Once host IDs are used in NetPoint, you must list every possible way to address a host in the host IDs list, or the missing addresses (hostname/alias) can be used to gain non-protected access. NetPoint's preferred host feature uses host ID lists to prevent the inadvertent creation of security holes.

Configuring a preferred host forces WebGate to pass the preferred host string to the Access Server for policy evaluation instead of the host typed into the browser by the user. No matter what is typed into the browser, the Access Server always sees the preferred host.

Denying Access to All Resources by Default

Access System default behavior is to allow access when a resource is not protected by a rule or policy. This is accomplished using one Boolean flag, DenyOnNotProtected, located in the *WebGate_install_dir*/access/oblix/apps/webgate/WebGateStatic.lst file. The default setting is false, which means that access is *allowed* to resources not protected by a NetPoint rule or policy. This parameter can be set to true.

When set to true, NetPoint's DenyOnNotProtected parameter lets you establish the opposite behavior. When set to true, DenyOnNotProtected denies access to all resources to which access is not explicitly allowed by a rule or policy.

Important: DenyOnNotProtected overrides Host Identifiers and Preferred Host. Leaving DenyOnNotProtected set to false can cause security holes in large installations with multiple Host Identifiers, virtual hosts, and other complex configurations.

Because different Web servers and Access Clients have different requirements, DenyOnNotProtected is implemented through WebGate. DenyOnNotProtected *cannot* be used with other types of AccessGates.

To deny access to all unprotected resources

1. Locate the *WebGate_install_dir*/access/oblix/apps/webgate/WebGateStatic.lst file.
2. Change the DenyOnNotProtected setting to true to *deny* access to all unprotected resources.

   Note: If you have set this parameter to true, you must protect Login.html with the "None" authentication scheme; otherwise, the page will not display when you access its associated resource.

   ---

3. Restart the WebGate to enable the change to take effect immediately.

Example of Using DenyOnNotProtected

Suppose you have a machine with IP addresses A and B associated with it, both on port 80, and using the same configuration file. For Netscape/iPlanet, this would be the obj.conf files. Both of these virtual servers are protected by the same Access Client or WebGate.

The goal is to protect all content on both virtual servers without using a Preferred Host. To meet this goal, you may set up a host ID for all variations of A, and then protect some content on A by defining policies for specific URLs. You need not set a Preferred Host for either Access Client A or B. You may also set the value of DenyOnNotProtected to true for the WebGate protecting the Access Client, so by default all content is protected on A and B.

With this setup, when a user tries to access a URL on A, the policies are evaluated first and if no corresponding Access Policy is found, content is denied only for A.

The Access Login Process

When a user or an application, such as a .JSP or Java application, attempts to access a NetPoint application or a NetPoint-protected Web resource, the login process is set in motion. This process varies depending on factors such as:

- **Is the Resource Protected or Unprotected?** If the resource is protected, what is the type of authentication scheme used?

  If the resource is protected by a WebGate, NetPoint challenges the user as specified by the challenge method configured in the authentication scheme.

  If the resource is an unprotected COREid application such as the User Manager, the COREid System uses its own login form to challenge the user for credentials.

- **Is the User Really Who the Person Claims to Be?**

  To ensure that the user is really who the person claims to be, WebGate challenges the user for credentials. If the credentials match, the WebGate authenticates the user, generates the ObSSOCookie, and sets it in the user's browser. WebGate generates the ObSSOCookie regardless of whether the authentication is successful or not.

  For information on the ObSSOCookie, see "Cookies Generated During Login" on page 91.

- **Has NetPoint Single Sign-On (SSO) Been Set Up?**

  NetPoint's SSO capability enables users to access more than one protected URL or application with a single login. If NetPoint SSO has been implemented in a single domain, the user needs to authenticate only once to access multiple resources protected by an authentication scheme which has the same level or a lower level of security. The ObSSOCookie is passed from the user's browser to any WebGates configured for the domain.

  When NetPoint SSO is implemented in a multi-domain environment, an authentication is honored by all the hosts in two or more domains.

  See the "Configuring Single Sign-On" on page 295 for more information on configuring SSO for single- and multi-domain environments.

- **Is the user allowed to access the content? If so, what actions can the person perform?**

  WebGate queries the Access Server to determine whether the user is authorized to access the resource. Access Server checks whether the user is authorized to access the resource. If the user is authorized, the Access Server checks for a policy that specifies the actions that the user is allowed perform.

  The Access Server sends the information to WebGate, which then returns the requested resource to the user.

Figure 1 illustrates the NetPoint authentication process.

Figure 1 The NetPoint Authentication Process

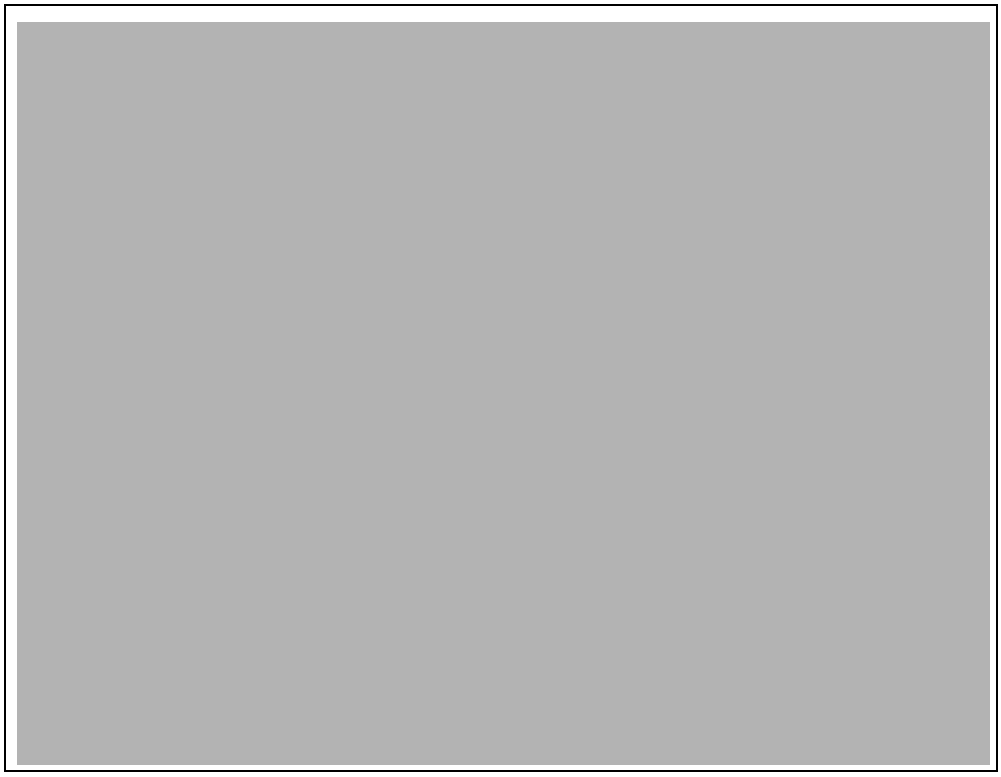
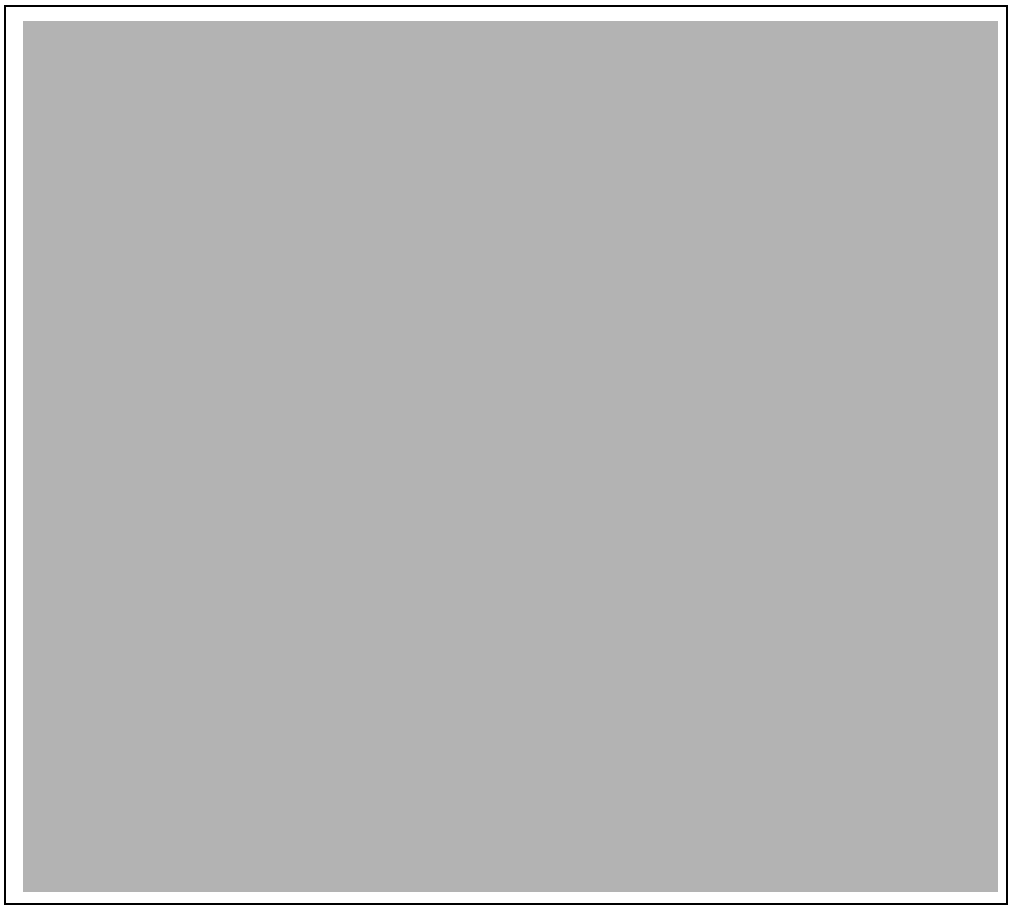


Figure 2 illustrates the authorization process.

Figure 2 illustrates the NetPoint authorization process:

Figure 2 The NetPoint Authorization Process



Login Processes

This section describes different scenarios where a user attempts to access a NetPoint-protected resource.

Process overview: Access when COREid is not protected by WebGate

1. A user attempts to access a COREid System application that is not protected by a WebGate.
2. The NetPoint application challenges the user for credentials such as user name and password.

3. If the user authenticates successfully, the COREid application generates the ObTEMC and the ObTEMP cookies.

   See “Cookies Generated During Login” on page 91 for information on cookies.

4. NetPoint then allows the user to access the COREid application. The user can perform specific actions in accordance with how the application is configured.

Process overview: Access when the resource is protected by WebGate

1. A user attempts to access a resource that is protected by a WebGate.

   When the user attempts to access a Web resource or an application, WebGate intercepts the request and queries its cache to determine if the requested resource and the associated operations are protected.

2. If information on the requested resource is not in the cache, WebGate makes a request to the Access Server for the security policy to determine if the resource is protected by NetPoint.

3. If the resource is an unprotected NetPoint resource, WebGate forwards the request to the server storing the resource, and the NetPoint application authenticates the user as in the previous process overview on page 88.

   If the resource is protected, WebGate looks for the ObSSOCookie to determine whether the user has already been authenticated.

4. If the user is not authenticated, the server challenges the user for credentials. The challenge method varies depending on the authentication scheme used.

   If a form-based authentication scheme is used, WebGate generates the ObFormLoginCookie. See “Cookies Generated During Login” on page 91 for information on the ObFormLoginCookie.

5. If the user authenticates successfully, WebGate generates the ObSSOCookie and sets it for inclusion in the next response to the user’s browser.

   Depending on the actions specified for authentication success and authentication failure, the user may be redirected to a specific URL, or user information may be passed on to other applications through a header variable or a cookie value.

6. WebGate then queries the Access Server for information on authorization for the resource.

7. The Access Server queries and evaluates the appropriate authorization policies stored in the directory server, and passes on the information to WebGate.

   For SSO to a NetPoint application, the authorization policy must set an action to set the HTTP_OBLIX_UID header to the user identity for the NetPoint application. If this header is not set, the application authenticates the user as in the previous process overview on page 88.

8. If the user is authorized, access to the requested content is allowed, and the HTTP_OBLIX_UID header is set.

   Depending on the authorization actions specified for authorization success and authorization failure, the user can be redirected to a specific URL, or user information can be passed on other applications through a header variable or a cookie value.

9. The NetPoint application reads the HTTP_OBLIX_UID header variable to get the identity. The NetPoint application determines the user's access rights from the identity.

Process overview: COREid resource protected by WebGate

1. A client application, such as a .jsp or a Java application, attempts to access a URL to a COREid resource that is protected by WebGate.

   The client application uses Access Server API to interface with NetPoint's Authentication and Authorization services. The application supplies the client's user credentials to the Access Server API.

2. The Access Server queries and evaluates the appropriate authentication rule.

3. The Access Server API authenticates the user and creates a session token.

4. The Web application sets the ObSSOCookie with the session token for the domain containing the client application and sends the cookie along with the request to the client application.

5. The WebGate queries the Access Server for information on authorization for the client user.

6. The Access Server queries and evaluates the appropriate authorization policies stored in the directory server, and passes on the information to WebGate.

   For SSO to a NetPoint application, the authorization policy must set an action to set the HTTP_OBLIX_UID header. The header must set the user identity to be used by the NetPoint application. If this header is not set, the application authenticates the client application itself, as in the process overview on page 88.

7. If the client application is authorized, access to the requested content is allowed and the HTTP_OBLIX_UID header is set.

   Depending on the authorization actions specified for authorization success and authorization failure, the user may be redirected to a specific URL, or user information may be passed on other applications through a header variable or a cookie value.

Note: If the client application is authorized, it is allowed to access the resource.

Cookies Generated During Login

Depending on the scenario, NetPoint generates one or more cookies. Cookies contain information such as the user DN, the client's IP address, and the cookie expiry time.

- ObSSOCookie
- ObFormLoginCookie
- ObTEMC Cookie
- ObTEMP Cookie
- ObPERM Cookie

ObSSOCookie

The ObSSOCookie is an encrypted single sign-on cookie that is generated by WebGate when a user authenticates successfully. The ObSSOCookie, a session-based cookie, stores user identity information. You can cache the information, if necessary.

See "Configuring Single Sign-On" on page 295 for more information on the ObSSOCookie.

ObFormLoginCookie

The ObFormLoginCookie is generated when a NetPoint form-based authentication scheme is used to protect a Web resource. WebGate uses the ObFormLoginCookie to direct the user back to the requested resource after successful authentication.

The ObFormLoginCookie maintains the original request information. By default, this cookie is set when the browser is first redirected to the form. The ObFormLoginCookie contains the following information for the original request:

- The requested URL
- The requested operation
- An authentication scheme
- The host to return to URL

See "Form-Based Authentication" on page 349 for more information.

ObTEMC cookie

The ObTEMC cookie, an encrypted session-based cookie, is generated by the COREid application when a user authenticates successfully. The ObTEMC cookie contains the following information:

- User distinguished name (DN) and the original DN. Original DN information is stored only if the COREid Substitute Rights feature is used. For details about adding substitute administrators and substitution rights, see *Volume 1*.
- A flag specifying whether the user is a NetPoint Admin, an Identity Administrator, or an Access Administrator.
- If single sign-on (SSO) has been implemented, the SSO Login ID.
- The time stamp.

  Every time a user performs an action, the time stamp is updated in the cookie to reflect the last time the session was used. If SSO has been implemented, however, the time stamp is ignored.
- The IP address of the client machine.

ObTEMP cookie

The ObTEMP cookie is a session-based cookie that is generated by the COREid application when a user authenticates successfully. The ObTEMP cookie contains the following application information:

- Login name
- User type
- Number of search-generated results (selector info)
- Uncommitted changes in various configuration applets

ObPERM cookie

The ObPERM cookie is a permanent cookie that is stored on the client machine. Between user sessions, the ObPERM cookie stores the following application information:

- Application style
- Custom view search results

Section II: Protecting Resources

3 Protecting Resources with Policy Domains

The NetPoint Access System enables you to protect your resources with policy domains. Policy domains include authentication and authorization rules determining who can access the resources. You can also create policies within a policy domain to define finer-grained protection for resources. For each policy domain and policy, you can define audit rules to monitor and record events, including system events, successful and failed user authentications, and successful and failed authorization of users who request access to protected resources.

This chapter provides an overview of policy domains. It explains how to create policy domains and policies. It also explains auditing and how to configure it. This chapter discusses the following topics:

Prerequisite Tasks

NetPoint 7.0 should be installed and running properly, as described in the *NetPoint 7.0 Installation Guide*.

Before you read this chapter, be sure to read the following:

- "Configuring AccessGates and Access Servers" on page 33, which describes how to configure AccessGates and Access Servers, something you must do before the policy domains you create can take effect.
- "Configuring User Authentication" on page 149, for information describing authentication and how to configure and manage it.
- "Configuring User Authorization" on page 229, for information describing authorization and how to configure and manage it.

In addition, the NetPoint Administrator must complete the tasks described below before any policy domains can be created.

Task overview: Prerequisite tasks for a NetPoint Administrator

1. Define the policy base during Access Manager setup, as described in "About the Policy Base" on page 97.

   To review the policy base from the Access System Console > System Configuration > View Server Settings page, look for the Policy Data Configuration section to obtain machine name, port, root DN, directory server security, and policy base.

2. Define the policy domain root during Access Manager setup, as described in "About the Policy Domain Root" on page 97.

3. Create the Master Access Administrator who has the authority to create policy domains, resource types, access control templates called schemes, and to assign other administrators the role of Delegated Administrator of a policy domain:

   - Master Access Administrators can be created after installation, as described in "Configuring Master Access Administrators" on page 23.
   - Master Access Administrators can delegate their authority, as described in "Delegating Policy Domain Administration" on page 145.

About the Policy Base

NetPoint must maintain information about the policy domains you create. NetPoint policy data includes the rules that govern access to resources.

During Access Manager installation, the NetPoint Administrator is asked to specify where NetPoint policy data will be stored. During Access Manager setup, a policy base is requested. The policy base is the location in the NetPoint LDAP directory that is the beginning point—the base—where NetPoint stores Oblix-specific object classes for policy domains and their policies.

All information regarding the definition of a policy domain is stored in relation to the policy base. The definition of a policy domain—its rules and the identification of its resources—is stored at the same level under the policy base.

The policy base must be defined before performing any tasks pertaining to policy domains or policies. See the *NetPoint 7.0 Installation Guide* for details.

About the Policy Domain Root

During Access Manager setup, the NetPoint Administrator is asked to specify a policy domain root. This is the first URL prefix for a policy domain under which all resources are protected. The default policy domain root must be broad to provide a wide scope that encompasses all of your resources. The default root is /.

For details about URLs, see “Configuring URLs for Resources” on page 111. For information describing how the policy domain root was created during Access Manager setup, see the *NetPoint 7.0 Installation Guide*.

About Policy Domain Administration

To protect resources, you create *policy domains*. A policy domain consists of:

- The resources you want to protect

  You can protect Web-based and non Web-based resources, including Web pages, site domains (collections of Web resources), servers, files, applications, and other executable programs.

  To include resources in a policy domain, you specify a URL that includes the resource at some level. You can specify the URL of a single resource—such as a large-scale application—to protect it, or you can protect entire directories of folders, files, and executables under a URL.

- Authentication, authorization, and auditing rules
- An authorization expression

- Policies to protect subsets of resources within the policy domain.

  A policy consists of a set of resources, authentication, authorization, auditing rules, and an authorization expression.

- The rights given to various administrators to create and modify the policy domain

---

Note: Before you can protect a resource, the NetPoint Administrator must define the policy domain root and policy base, as described in “Prerequisite Tasks” on page 96, and define a Master Access Administrator as described in “Configuring Master Access Administrators” on page 23.

About Creating the First Policy Domain

A Master Access Administrator must create the first policy domain after the policy domain root is defined. He or she can then create policy domains for URLs beneath the first one and delegate administration of those policy domains to other administrators. For details about the policy domain root, see “About the Policy Domain Root” on page 97.

Task overview: Creating the first policy domain

1. Define the resource types for any resources to be included in the domain whose types NetPoint has not already defined by default. See “Configuring Resource Types” on page 107.
2. Create the Master Audit Rule. See “Configuring the Master Audit Rule” on page 132.
3. Create the Authentication Scheme for the policy domain. See “Authentication Schemes” on page 152 and “Authentication Rules” on page 205.
4. Create the Authorization Scheme for the policy domain. Refer to either of the following sections:
   - “Configuring Authorization Rules” on page 238, which includes information about using a NetPoint-provided authorization scheme
   - “Authorization Schemes for Custom Plug-Ins” on page 287
5. Configure the URLs for the resources of the first policy domain. See “Configuring Resource Types” on page 107.
6. Create the Authentication Rule for the policy domain. See “Authentication Rules” on page 205.
7. Create actions for the Authentication Rule. See “Authentication Actions” on page 211.
8. Create one or more Authorization Rules. See “Authorization Rules” on page 233.

9. Create actions for the Authorization Rules. See “Setting Actions for Authorization Rules” on page 281.
10. Create an authorization expression for the policy domain containing one or more authorization rules. See “Authorization Expressions” on page 247.
11. Create the audit rule for the policy domain. See “Creating an Audit Rule for a Policy Domain” on page 140.
12. Test the policy domain. See “Using Access Tester” on page 143.
13. Delegate management of the domain to a Delegated Access Administrator. See “Delegating Policy Domain Administration” on page 145.

About Managing a Policy Domain

As Delegated Access Administrator, you can manage a policy domain for which you have been granted administrative responsibilities and privileges.

The tasks you can perform to administer a policy domain for which you have administrative rights are listed below. You can perform these tasks as needed; none is required.

Task overview: Administering a policy domain

1. Replace the existing authentication rule for the policy domain or its policies. You must first delete the policy domain’s or the policy’s existing authentication rule.

   To create an authentication rule, you select an authentication scheme created by a Master Access Administrator, and configure the rule’s actions. See “Authentication Rules” on page 205.
2. Replace the existing authorization expression for the policy domain or its policies. You must first delete the content of the policy domain’s or the policy’s existing authorization expression.

   To create an authorization expression for the policy domain or any of its policies, you combine one or more authorization rules created at the policy domain level. See “Authorization Expressions” on page 247.
3. Create audit rules derived from the Master Audit Rule. See “Creating an Audit Rule for a Policy Domain” on page 140 and “Defining an Audit Rule for a Policy” on page 141.
4. Test the policy domain after making changes to it. See “Using Access Tester” on page 143.

Overview for Delegated Access Administrators Creating a Policy Domain

As a Delegated Access Administrator with certain rights, you can create policy domains. See “Types of administrators and their rights.” on page 146.

You can create policy domains to include resources whose URLs are already covered by the ones included in a policy domain for which you have administrative privileges and responsibilities. For details, see “How the Policy Domain or Policy for a Resource Is Determined” on page 104.

Task overview: Creating a policy domain

1. Configure the URLs for the resources of the policy domain. See “Configuring URLs for Resources” on page 111.
2. Create the authentication rule for the policy domain. See “Authentication Rules” on page 205.

   You include in the rule an authentication scheme created by a Master Access Administrator.
3. Create actions for the authentication rule. See “Authentication Actions” on page 211.
4. Create one or more authorization rules for the policy domain. See “Authorization Rules” on page 233.
5. Define actions to be taken for the authorization rule, if the rule fails or if it succeeds. See “Authorization Actions” on page 276.
6. Create the authorization expression for the policy domain containing one or more authorization rules. See “Authorization Expressions” on page 247.
7. Create actions to be taken for the authorization expression, depending on the evaluation of the expression: success, failure, or inconclusive. See “About Actions For Rules and Expressions” on page 277.
8. Create the audit rule for the policy domain. See “Creating an Audit Rule for a Policy Domain” on page 140.
9. Test the policy domain. See “Using Access Tester” on page 143.
10. Configure policies for the policy domain, if any. See “Configuring Policies” on page 136.

About Policy Domains and Their Policies

Policy domains are logical structures defined for resources you want to protect in the same way. To provide different and more specific coverage to a subset of resources within the domain, a policy domain can contain policies.

Users request access to resources protected by a policy domain, and their requests are assessed according to the domain's authentication rule and its authorization expression.

There are a number of ways users can attempt to access a resource protected by a policy domain: for example, by entering the URL for a resource in a browser, attempting to execute an application, or calling some other external business logic.

Parts of a Policy Domain

A policy domain consists of the following parts:

- URLs that define paths covering resources protected by the domain's authentication rule and authorization expression

  A policy domain can include multiple URLs which are independent of one another. Resources under one URL might reside on a host different from resources under another URL belonging to the same policy domain. The policy domain's default rules apply to the resources it contains, unless the resource is protected by a specific policy.

- The host identifier

  All resources added to a policy domain are identified by the host on which they reside and their URLs. A host can be known by multiple names. To ensure that it recognizes the URL for a resource, NetPoint must know the various ways used to refer to that resource's host machine.

  The Host Identifiers feature allows you to enter the official name for the host and every other name by which the host can be addressed by users. A request sent to *any* address on the list is mapped to the official host name. For details about the host identifier, see "Configuring AccessGates and Access Servers" on page 33.

  It is possible to use the Host Identifiers feature to set up a host context for adding resources to the same policy domain on different machines. For details describing what a host context is and why you may want to use one, see "Using Host Identifiers and Host Contexts" on page 127.

- Rules and expressions for protection

  Rules for authentication determine how the identity of a user attempting to access a resource is to be proven. Authentication rules contain authentication schemes. Authorization rules determine whether the user has the right to access the resource. Authorization rules contain authorization schemes and are contained in authorization expressions. An authorization expression can contain one or more authorization rules. Auditing rules determine the information to be recorded in the audit log for operations pertaining to the policy domain or policy (audit). Auditing rules are derived from a Master Audit Rule. For details, see the following information:

  - For details about authentication rules and authentication schemes, see “Configuring User Authentication” on page 149.
  - For details about authorization schemes, authorization rules, and authorization expressions, see “Configuring User Authorization” on page 229.
  - For details about auditing rules and the Master Audit Rule, see “About Rules and Expressions” on page 119 and “Configuring the Master Audit Rule” on page 132.

- Policies for URL patterns and the operations allowed for the type of resource to which the pattern applies

  Policies for resources within a policy domain allow you to create finer-grained ways to protect specific resources in the domain. You can specify a URL pattern or an explicit URL to identify resources. Different types of resources have their own operations. You can specify the operations—also known as request methods—that are allowed for resources of a type. Requests for resources whose URLs match the pattern are further processed against the rules of the policy.

  For details about policies, see “Configuring Policies” on page 136.

Figure 3 on page 103 provides a conceptual view of the parts of a policy domain. In this figure, only Web-based resources are shown. However, NetPoint Access System policy domains can also protect resources other than Web-based ones.

Figure 3 Policy Domain Structure

A policy domain can contain different types of resources, such as:

- An entire external Web site
- Specific pages in a Web site
- Partner portals
- A parts order application
- Invoice applications
- A benefits enrollment application on Web servers of an enterprise in many countries

For details describing resource types, see “Configuring Resource Types” on page 107.

How the Policy Domain or Policy for a Resource Is Determined

A resource may fit within the definitions of more than one policy domain. It may fall within a broadly defined policy domain such as /mydomain. It may also fall within a more specific policy domain such as /mydomain/myresources. The policy domain that a resource belongs to is always the more specific one for the resource's URL. The NetPoint Access Server checks all policy domain definitions to find the policy domain having the most specific URL prefix matching the resource.

The Access Server checks policies in the order you specified when you configured them. It uses the first matching policy regardless of how many more policies there are. It may be that a policy that was not checked has a more specific URL matching the requested resource. In this case, the policy would not be checked because a previous policy provided the match. For the intended policy to be used and for processing efficiency, you should consider the order you assign to policies.

Preconfigured Policy Domains Provided by NetPoint

You can install policy domains to provide protection for NetPoint CoreID and Access Manager resources (URLs). If the policy domains were created during installation, you must configure a WebGate and Access Server for them, and then enable or disable them together. The policy domains created during installation are:

- **NetPoint Identity Domain**—Protects NetPoint Identity URLs.
- **NetPoint Access Manager**—Protects NetPoint Access Manager URLs.

See the *NetPoint 7.0 Installation Guide* for information about configuration of authentication schemes and policies during the installation process.

Who Creates Policy Domains?

You can distribute policy domain creation and administration across your organization and down through it to the various administrators responsible for management of resources.

You can centralize policy creation while decentralizing management and enforcement of it. You may want a Master Access Administrator to create several policy domains and delegate administration of them to various Delegated Access Administrators. The Delegated Access Administrators can manage the domains and create policy domains for resources whose URLs are more specific than those of the domain.

Examples

- A Web Master may be responsible for maintenance of a corporate Web site. The Web Master is assigned the position of Delegated Access Administrator

for the policy domain for this resource. The policy domain includes other resources the Web Master also manages.

- A Delegated Access Administrator may manage a particular resource, such as a powerful, feature-rich application used internationally throughout an enterprise. One such resource might be Arete Airline's passenger check-in verification system. Instances of the application may run on many servers.

  Because a related, smaller application called Upgrade requires the same protection and is managed by the same administrator, both applications could belong to the same policy domain. Additionally, all instances of each application could be protected by the same policy domain.

Delegation of management of policy domains enables you to scale administration of your resources empowering those closest to the resources and most knowledgeable about them to manage them. For details about Master Access Administrators and Delegated Access Administrators and their rights and responsibilities, see "Delegating Policy Domain Administration" on page 145.

Policy Domain and Policy Examples

Organizations use policy domains and policies for different purposes and in different ways. An organization's design for use of policy domains and policies is unique to it.

Here are some example scenarios that entail use policy of domains:

Policy Domain for Human Resources and Marketing—A policy domain protects human resources information made public to employees and a branch of the marketing Web site. Both sets of resources require the same kind of protection.

The following two URLs define the policy domain's logical structure:

/AreteAirlines/marketing/reports/

/AreteAirlines/HR/

If resources of either organization in the policy domain require protection rules more specific than the policy domain's default rules, policies can be used to protect those resources. For example:

- The default Authorization rule for /AreteAirlines/HR/ grants all users weekday access only. A policy could be used to remove weekend access restrictions from a set of human resource files for human resources management personnel, who tend to work weekends.

- The same policy domain includes resources in a private directory to be viewed only by regular employees, such as analysts' reports.

  The private directory is subordinate to the reports directory. Resources in the private directory are protected by the default rules of the policy domain unless a policy is used to provide them different protection. A policy that restricts access to the resources in the private directory exists; it stipulates that only regular full-time employees may see the reports in the following private directory:

  /AreteAirlines/marketing/reports/private/

- The policy domain's URLs encompass a resource that is an application called badgit. The application enables HR employees to register employees of the organization for access badges. The main application processes the request and obtains information from a backend application. A policy is used to protect only this application. The policy applies to the following specific URL:

  /AreteAirlines/HR/badges/badgit.exe

Policy Domain for a University—A university provides information to its students, but not to outsiders. The URL for the policy domain protecting the resources is:

/GlobalUniv/

Two policy domains with more specific URLs are created to include resources otherwise covered by the /GlobalUniv/ policy domain.

- One of these policy domains includes the URL /GlobalUniv/physics/.
- The other policy domain includes the URL /GlobalUniv/philosophy/.

The policy domain /GlobalUniv/physics/ allows all students of the university to access the policy domain's resources.

- All students—physics students, philosophy students, and any others—can access resources in the /GlobalUniv/physics/feynman/diagrams/ directory because the default rules of the /GlobalUniv/physics/ policy domain apply, and there are no specific policies applied to these resources.

- A policy is created to allow only those students who meet the authorization criteria of the policy protecting the testResults.html page to see it. The students who took a quiz may be able to view the following Web page:

  /GlobalUniv/physics/feynman/diagrams/testResults.html

The college presents a suite of applications animating the world of black holes. The applications are available to all students, not just physics students. The URL for one of these Enterprise JavaBean (EJB) applications is /GlobalUniv/physics/wheeler/blackHoles/explore/styx.ejb.

Because the application is in a directory called wheeler, which is subordinate to the physics directory, a policy must be used to remove access to the wheeler directory, uncovering the resources for all science students.

About Allocating Responsibility for a Policy Domain

You can assign to various users administrative roles and give them the privileges and responsibilities for managing policy domains for resources of the same or a different type on the same or a different host. It is a good idea to define policy domains along the lines of the resources they protect and who manages them. Who can access them is secondary, and is expressed through the access control rules of a policy domain. How you design and implement policy domains is determined by the requirements of your organization.

Here are some examples of reasons why decentralizing management of resources may be useful:

- You want to provide your employees with faster and better service for your online applications. For example, improved service helps to make applications more readily available initially and more easily recoverable if a failure of the host system occurs.
- You may want to keep your informational Web sites for employees operational and current with as little disturbance as possible.

Configuring Resource Types

A resource type describes the kind of resource to be protected, including its associated operations. Operations associated with a resource are tied to its type.

Before you can add resources to a policy domain, you must define their types and the operations associated with them that you want to protect.

The NetPoint Access System defines some default resource types. If you want to protect types of resources different from the default ones, you must define their types. Only the Master Access Administrator can create resource types. Resource types are created from the Access System Console.

By giving you the ability to define more resource types than the default ones provided, NetPoint enables you to protect more than just Web-based resources.

Resource Types Defined by NetPoint

By default, the NetPoint Access System defines resource types for Enterprise Java Beans (EJB) and HTTP (HTTPS) resources. The HTTP resource type definition is required, whether or not you protect resources of this type. You cannot delete the HTTP resource type or modify its operations. The EJB resource type is not required. You can delete it if you do not plan to protect EJB resources. You must define the type for any other types of resources that you want to protect.

Supported HTTP Operations

NetPoint supports the following HTTP operations:

- **CONNECT**—Handshakes with a URL
- **DELETE**—Deletes information from the URL, or deletes the URL itself
- **GET**—Retrieves information from the URL
- **HEAD**—Obtains information about the resource without making changes to the URL
- **OPTIONS**—Obtains information about HTTP methods available to and from the URL
- **OTHER**—Non-standard, custom operation
- **POST**—Copies information to the URL
- **PUT**—Replaces a file or document in the URL
- **TRACE**—Views information about what the URL is receiving

Supported EJB Operation

NetPoint supports the EJB EXECUTE operation, which executes a bean. You can add other EJB operations.

Supported Resource Types

A policy domain can protect these types of resources and their operations:

- EJB Resources

  EXECUTE

- HTTP Resources

  GET, POST, PUT, TRACE, HEAD, CONNECT, OPTIONS, and others

  You can define policies to protect a specific operation.

- RDBMS Resources

  ADD, DELETE, and UPDATE

- Servlet resource types

The following table shows examples of HTTP resources, Java 2 Enterprise Edition (J2EE) resources, and other online application resources identified by their URLs.

Resource Type	Examples
HTTP Resources	• Directories /mydirectory • Pages /mydirectory/index.html • Web applications /applications/myexe.exe • Query strings www.wwm.com/sales/result/pricelist1,2,0-a-00,000.htm?st.dl.search.qs.results
J2EE Application Server Resources	• Java Server Pages JSP(s) • Servlets • Enterprise Java Beans
Other Resources	• Standalone programs Java, C, C++ application programs • ERP applications • CRM applications

Defining a Resource Type

To define a resource type, use the Define a New Resource Type page.

To define a resource type

1. From the Access System Console click the Access System Configuration tab, then click Common Information Configuration > Resource Type Definitions.

   The List All Resource Types page appears.

2. On the List All Resource Types page, click Add.

   The Define a new Resource Type page appears.

3. In the Resource Name field, enter a unique name for the new resource type.
4. In the Display Name field, enter the name of the resource type.
5. In the Resource Matching field, specify whether the resource type can be read as case sensitive or case insensitive.
6. In the Resource Operation field, specify the operations this resource type can perform.

   You can define custom operations, but not for HTTP resource types.

   To add or delete fields as necessary, click the plus (+) and minus (–) icons.

7. Click Save to save your changes (or click Cancel to exit the page without saving).

Configuring URLs for Resources

To use the NetPoint Access System to protect your resources—for example, your business applications and content—you must create a policy domain whose URL prefixes and URL patterns identify the resources.

URL Prefixes—You use URL prefixes to define the policy domain content. For policies, you use URL patterns to identify resources protected by the policy.

You can create URL prefixes that define a broad scope of content, for example:

/

/sales

/humanresources

In this example of a policy domain, the resources to be protected exist on three different hosts. All resources under the URL prefixes are protected by the default rules of the policy domain:

Policies—You can create policies for resources within a policy domain. For example, resources for two other groups reside under /. They are engineering and marketing.

- Because no policy is defined for /engineering, its resources are still protected by the default rules of the policy domain. Default rules also apply to marketing.
- After the administrator creates a policy for resources under /engineering, the engineering resources are protected by the rules specified by the policy and not the default rules of the policy domain.

URL Patterns—You can create policies with granular URL patterns. Here is an example of a URL pattern:

/.../update.html

This URL pattern matches these resources:

/humanresources/benefits/update.html

/corporate/news/update.html

update.html

Figure 4 illustrates how URL prefixes and URL patterns are used to define the resources for policy domains and their policies

Figure 4 URL Prefixes and Patterns

URL Prefixes

The URL prefix is the starting point for resources in a policy domain. A URL prefix defines the beginning boundary of a policy domain, that is, its first resource. A URL prefix maps to a directory on the file system of one of your application servers or Web servers.

All resources under the URL prefix are protected by the default rules of the policy domain unless more specific rules are applied to them through policies. You can assign one or more URL prefixes to a policy domain, but each URL prefix can belong to one policy domain only.

The trade-off in creating many granular policy domains to protect your resources is that you achieve greater security at the cost of increased overhead. The cost is incurred because the Access Server must evaluate all policy domains to find the one that is most specific to the resource. Use of policies affords you the same benefit without the overhead.

Process overview: How a URL prefix is used

1. An end user requests a resource by specifying the URL for the resource. A user enters the following URL in her browser to request access to a data page displaying information about a specific corporate partner:

   www.AreteAirlines.com/Partners/mycorp.html

   If the user's own Web site is set up accordingly, the user may select a link which represents the resource (and the URL for it).

2. The NetPoint Access System locates the requested resource. The Access Server assesses all of the policy domains to ascertain the one having the URL prefix most specific to the incoming URL for the resource. (The Access Server determines if the resource is covered by a policy within the domain, whose rules would then apply.)
3. If no policy applies, the Access System uses the rules of the policy domain to determine whether to allow or deny the user access to the HTML page.

A policy domain can protect content other than Web-based content, although the policy domain in Figure 4 on page 112 covers Web-based resources.

You can specify individual policies for resources of a given type whose URLs match a URL pattern. You can also specify the kinds of operations that can be performed on the resources.

A URL pattern is a NetPoint-supported mechanism for identifying different resources of a certain type protected by a single policy. A URL pattern can be a directory, query string pattern, or query string variable. If it is an explicit fully qualified URL, then it refers to a single resource.

An example of a URL pattern covering many resources is a URL for all HTML pages (*.html) of a department's Web site. In this case, the policy may remove restrictions imposed by the policy domain's default rules. An example of a URL pattern for a specific file is an explicit fully qualified path (URL) of a single instance of an application. Resource operations are the functions available for each configured type of resource. For example, HTTP has GET, POST, PUT and other operations.

How URL Patterns are Used

URLs for policies specify the fine-grained portion of a resource's namespace. To fully identify the URL, the host identifier and URL prefix for the policy domain are concatenated with the policy's URL pattern.

Process overview: How URL patterns are used

1. A user specifies the URL for a requested resource.
2. Based on the policy domain's host and URL prefix information, Access Server creates a fully qualified URL that includes the URL pattern.
3. The Access Server compares the incoming URL for the requested resource to the fully qualified URL constructed from the policy domain information and the policy's URL pattern.
   - If there is a match, the policy's various rules are evaluated to determine whether the requester should be allowed or denied access to the resource.
   - If requester is allowed access, the resource is served to the user.

Figure 3 on page 103 shows the structure of a policy domain called Partners that includes the following URL pattern:

/Ace/.../*

To get the fully qualified name of the URL pattern for the policy, the policy domain's URL prefix, /Partners, is prepended. The name of the host where the resources of the policy domain reside is specified before the URL prefix, resulting in the following URL:

myhost/Partner/Ace/.../*

URL Pattern Matching Symbols

NetPoint expresses URL patterns through *globbing*, which is like filtering. NetPoint's globbing combines different Unix shell (sh, csh or tcsh) support for patterns in file names with NetPoint-provided patterns such as "..." (three periods), which let you span multiple directories.

Table 4 shows NetPoint supported patterns.

Table 4 NetPoint supported patterns

Pattern	Description	Examples
?	Matches any one character other than /.	a?b matches aab and azb but not a/b.
*	Matches any sequence of zero or more characters. Does not match /.	a*b matches ab, azb, and azzzzzzb but not a/b.
[set]	Matches one from a set of characters. A set can be specified as a series of literal characters or as a range of characters. A range of characters is any two characters (including -) with a hyphen (-) between them. The forward slash character (/) is not a valid character to include in a set. A set of characters will not match / even if a range that includes / is specified.	• [nd] matches only n or d. • [m-x] matches any character between m and x, inclusive. • [--b] matches any character between - and b inclusive (except for /; see /usr/pub/ascii for order of punctuation characters). • [abf-n] matches a, b, and any character between f and n, inclusive. • [a-f-n] matches any character between a and f inclusive, -, or n. (The second - is interpreted literally because the f preceding it is already part of a range.)
{pattern1, pattern2,...}	Matches one from a set of patterns. The patterns inside the braces may themselves include any other special characters except for braces (sets of patterns may not be nested).	• a{ab,bc}b matches aabb and abcb. • a{x*y,y?x}b matches axyb, axabayb, ayaxb, etc.
/.../:	Matches any sequence of one or more characters that starts and ends with the forward slash character (/).	• The pattern /.../index.html matches: /index.html /oblix/index.html /oblix/sales/index.html index.html It does not match xyzindex.html or xyz/index.html. • /oblix/.../*.html matches: /oblix/index.html /oblix/sales/order.html, and so on

Table 4 NetPoint supported patterns

Pattern	Description	Examples
\	The backslash character is used to escape special characters. Any character preceded by a backslash matches itself.	• abc*d only matches abc*d • abc\\d only matches abc\d

Invalid Patterns

Patterns with the following attributes are invalid:

- A '[' without a closing ']'
- A '{' without a closing '}'
- Unescaped '{' inside {}
- Unescaped '/' inside [ ]

How Pattern Matching Works at Access Runtime

You can obtain information about this subject in the Oblix Knowledge Base. You need a user name and password to log in. See "Preface" on page 15 for details.

Access System Patterns

A policy can contain one or more of the following types of patterns. If multiple patterns are specified in one policy, they *all* must match to the incoming URL. If they do not, the policy does not apply to the URL.

This example uses the following incoming URL:

http://www.myserver.com/oblix/sales/index.html?user=J.Smith&dept=sales

The policy includes the following URL patterns:

- Pattern for the absolute path of the URL

  This pattern is the part of the URL that does not include the scheme (http) and host/domain (www.myserver.com), and that appears before a ? character. In this example, the absolute path is: /oblix/sales/index.html.

- Pattern for name value pairs in the URL

  A set of these pairs may be configured as a pattern. The pairs apply to query data that appears after the ? character in the URL—if the operation is GET. If the operation is POST, query data appears after the POST data. For a pair, name specifies a name value, not a pattern. The value element of the pair is configured as a pattern. For example:

Name	Pattern
user	*Smith
dept	*sales*

  If multiple name-value pairs are specified, they all must match the incoming URL. Therefore, the following URL does not match the pattern:

  http://www.myserver.com/oblix/sales/
  index.html?user=J.Smith&dept=engg

  The important difference between this pattern and the next one is that there is no priority to these name-value pairs. The following URL satisfies the pattern:

  http://www.myserver.com/oblix/sales/
  index.html?dept=sales&user=J.Smith (with reverse order of “dept” and “user”)

  This is important and useful because it is commonly difficult to control the order of name-value pairs in the GET/ POST query data.

- Pattern on the entire query string:

  This is useful if you want to enforce an order on the query string. For example, a pattern:

  user=*Smith*sales*

  matches the query string

  user=J.Smith&dept=sales

About Schemes

Schemes allow the Master Access Administrator to define methods to be used to authenticate users and to verify a user’s right to access a resource. Schemes are reusable templates.

An authentication scheme contains one or more steps, each of which can include one or more plug-ins. Authentication schemes are contained in authentication rules.

An authorization scheme is included in an authorization rule, and one or more authorization rules are combined to form an authorization expression. You can use the default authorization scheme provided by NetPoint, or you can provide a custom one.

After a scheme is defined, Delegated Access Administrators of different policy domains can use the same scheme in rules for their domains or in rules for policies within their domains.

A policy domain must have at least one authentication rule and therefore one authentication scheme. A policy domain must have an authorization expression containing at least one authorization rule. Therefore, it must have at least one authorization scheme.

You can, at one time, define all of the schemes you and your Delegated Access Administrators will need for policy domains and policies, or you can define schemes as they are required. You create schemes in the Access System Configuration area of the Access System Console.

About Plug-Ins

Plug-ins are dynamically loaded shared libraries executed to perform authentication and authorization processes. They are contained in schemes, and they are used to request and process the information necessary to authenticate a user or authorize a user to access a resource.

Plug-ins perform specific tasks. For example:

- **For Authentication Schemes**—Authentication schemes contain one or more steps. It is the steps of an authentication scheme that contain its plug-ins. NetPoint provides default plug-ins which you can use, or you can provide custom ones. For example, every chained authentication scheme must have a plug-in which maps information obtained from the user—user credentials—to user profile information.

  Also, every authentication scheme includes a challenge method plug-in and a password verification plug-in. NetPoint provides plug-ins for these purposes, too. If you do not use the plug-in provided by NetPoint for this purpose, you must replace it with one that provides the same functionality.

  If you want to replace the plug-ins provided by NetPoint with custom ones, you must design your plug-ins to perform required tasks, to accept and pass required parameters, and to return defined function codes. For details, see “Plug-Ins for Authentication” on page 165.

- **For Authorization Schemes**—For authorization rules, you can use the default authorization scheme provided by NetPoint, or you can use a custom one. If

you want to use a custom authorization scheme, you must provide your own plug-ins for it. For details, see “Configuring User Authorization” on page 229.

You can create plug-ins for the following supported platforms:

- **MS Windows**—A dynamic link library (.dll) is used to implement shared libraries.
- **UNIX**—A shared library (.so) is used to implement shared libraries.

For information describing how to create custom plug-ins, see the *NetPoint 7.0 Developer’s Guide*.

About Rules and Expressions

Rules contain schemes that define how the resources of a policy domain are to be protected, including:

- How authentication of the user is to be performed.
- Whether a user has the right to access a domain resource and any conditions defining access rights. Authorization rules are included in authorization expressions. A policy domain must contain one—and only one—authorization expression.
- Events to be audited pertaining to the policy domain or policy.

Rules can include actions to be executed depending on the result of the evaluation of user information against the specifications of the rule.

A policy domain can include policies, which can contain their own rules and authorization expressions. Therefore, a policy domain can contain two levels of rules:

- Those that apply by default to all resources of the policy domain.
- Those that are part of a policy and apply to specific resources within the domain. These policy rules override the default rules of the policy domain for the resources they protect.

Authorization expressions include authorization rules and the operators used to combine them. You combine rules within expressions to create from simple to complex means of specifying who is allowed or denied access to the protected resources.

Authorization rules are reusable within a policy domain. You can use the same rule in an authorization expression more than once. Also, you can use the same rule in the expression for the policy domain and in expressions for any of its policies.

Table 5 defines the four types of rules for a policy domain or a policy.

Table 5 Types of Rules

Rule	Description
Authentication Rule	Specifies the method used to challenge and authenticate users requesting access to protected resources. Can specify actions to be taken if authentication is successful or if it fails. **Note:** Only one default authentication rule can be included in a policy domain. Each of its policies, however, can have its own authentication rule.
Authorization Rule	Allows or denies a user access to requested resources within a policy domain or policy. Can specify actions to be taken if authentication is successful or if it fails. These rules are included in authorization expressions. **Note:** If more than one rule is included in an expression, the order of evaluation of the rules is determined by the logic you specify to form the expression. Can also specify conditions for access. **Note:** Only one default authentication rule can be included in a policy domain. Each of its policies, however, can have its own authentication rule.
Authorization Expression	Includes authorization rules. A policy domain must have one and only one authorization expression. **Note:** A policy within a domain can have a single authorization expression. If it does not include one, the resources of the policy are protected by the authorization expression of the policy domain.
Audit Rule	Captures attributes and information about specific events pertaining to the policy domain. • It modifies and overrides events and information specified in the Master Audit Rule. • If no specific audit rules are applied, the Master Audit Rule is enforced by default.

Note: Authentication rules are applied before authorization rules because a user's identity must be proven before he or she is granted access to a resource.

Figure 5 illustrates a policy domain containing a default set of rules and a default authorization expression applied to the domain's resources. For the resources defined by the policy, the default rules and expression are overridden by those of the policy.

Figure 5 Rules and Authorization Expression for a Policy Domain and Policy

Lessening or Increasing Controls with Rules

By default, the Access System allows access to a resource that is not explicitly protected by a policy domain rule or a policy. You can begin to create policy domains from this condition—all resources unprotected. You can take the opposite position and reverse the default state so that all resources are protected at the outset.

Beginning with All Resources Unprotected

If you begin to create policy domains from a position in which all resources are unprotected, you must apply access controls to those resources. You can do this at a broad level by creating policy domains with default rules which are more or less restrictive:

- If you use restrictive default rules to impose tight controls across all resources of a domain, you can use policies to remove or change restrictions for subgroups of resources.
- If you use lenient default rules as a starting point, you can use policies to provide tighter, specific controls on subgroups of resources within a domain.

Beginning with All Resources Protected

To start from a state in which all resources are protected, you set the parameter DenyOnNotProtected. If this switch is set to true, DenyOnNotProtected denies access to all resources not explicitly allowed by a policy domain's rules or policies.

If all resources are protected, you must create policy domains and policies to remove protection from those resources you want to make available to various users. In this sense, you are uncovering resources to a greater or lesser degree to make them available.

You can do this at a broad level by providing default rules for a policy domain:

- If you use lenient default rules to lessen controls across all resources of a policy domain, you can use policies to apply particular restrictions for subgroups of resources.
- If you use tighter default rules as a starting point—perhaps rules that are stringent but less so than the current default state of complete denial of access—then you can use policies to lessen access control for subgroups of resources in various ways.

Note: If DenyOnNotProtected is set to false, this switch allows access to all resources not explicitly denied by a policy domain's rules or policies.

For information describing how to use the DenyOnNotProtected switch, see "Configuring AccessGates and Access Servers" on page 33.

Creating and Managing Policy Domains

This section describes how to create policy domains, enable or disable them, and manage their resources. It addresses the following set of tasks:

Creating a Policy Domain

Both Master Access Administrators and Delegated Access Administrators can create policy domains. Master Access Administrators can create policy domains at any level. Delegated Access Administrators can create policy domains that are subordinate to any policy domains delegated to them for administration.

You use the Access Manager to create policy domains, add resources to a domain, and protect the resources, using authentication rules and authorization rules and expressions.

Note: By default, a policy domain is not enabled by default. Do not enable a domain until you have added resources to it. Be aware that if you enable a policy domain that does not contain resources, the domain cannot be used.

To create a policy domain

1. From the Access Manager, click Create Policy Domain in the left side navigation bar.

   The Create Policy Domain page appears, as illustrated below.

2. In the Name field, enter a short alphanumeric string identifying the domain.

   NetPoint allows spaces to be used in this field.

3. In the Description field, type a brief description of this policy domain.

   The Name and Description appear in pages showing lists of policy domains. A description is optional.

4. Click Save.
   - To view currently defined information about your policy domain, click View as Page.
   - To return to the General page, click the name of your domain at the upper left part of the View as Page page.
5. When you are ready to *enable* a new policy domain, click Modify in the General page, select Yes in the Enabled field in the next page, then click Save.

   The General page reappears.

Modifying a Policy Domain

You can modify a policy domain after creating it. Modifying a policy domain includes changing any aspect of it— adding or removing resources, and modifying, removing, or adding rules.

Be sure to disable the policy domain before you modify it. See “Enabling and Disabling Policy Domains” on page 125 for details about enabling and disabling policy domains.

To modify a policy domain

1. From the Access Manager, select My Policy Domains.

   The My Policy domains page appears, displaying a list of policy domains.
2. Select the check box before the name of the policy domain you want to modify and click the domain’s name.
3. On the general page, click Modify at the bottom of the page.
4. Change any values you want to modify, and click Save.

Deleting a Policy Domain

You can delete a policy domain entirely without first removing its resources and rules. Before you delete a domain, disable it. See “Enabling and Disabling Policy Domains” on page 125 for details.

To delete a policy domain

1. From the Access Manager, select My Policy Domains.

   The My policy domains page appears, displaying a list of policy domains.
2. Select the check box before the name of the policy domain you want to delete.
3. Click Delete at the bottom of the page.

Enabling and Disabling Policy Domains

You must *enable* a policy domain before you can use it. You must *disable* a policy domain before you can modify its configuration.

Important: Disable a domain before modifying its rules or policies.

To enable a policy domain

1. From the Access Manager, select My Policy Domains.
2. In the My Policy Domains page, select the check box next to the domain you want to enable.
3. Click Enable.

   Yes appears in the Enabled column.

To disable a domain

1. From the Access Manager, select My Policy Domains.
2. In the My Policy Domains page, select the check box next to the domain you want to disable.
3. Click Disable.

   No appears in the Enabled column.

Searching for Policy Domains and Policies

You can search for and display existing policy domains and policies. Master Access Administrators can search for and see all policy domains and policies. Delegated Access Administrators can see only the policy domains for which they have been delegated administrative rights. For their policy domains, they can also see the policies which they have defined along with those defined by a Master Access Administrator.

You use the Search function to search for policy domains and policies.

To search for existing policy domains or policies

1. From the Access Manager, click Search.

   The Search window appears, as illustrated below.

2. In the left drop-down list of the Look For field, select either Policy Domain Name or Policy.
3. Select an entry from the drop-down list of search criteria in the middle, then type a text string in the right column.

   To find all entries that match the selected search criterion, leave the right column blank.
4. Click Start Search.

   The results display on your page.

Viewing General Information about Policy Domains

You can display a list of policy domains and view configured information for an individual domain. The My Policy Domains page displays a list of domains for which you have administrative rights. Master Access Administrators can see information about all policy domains. Delegated Access Administrators can see only the policy domains for which they have been delegated management.

To view policy domains and configuration information

1. From the Access Manager, click My Policy Domains in the side navigation bar.
2. Click the domain's link to view a domain's configuration settings.

   The General page displays the name and description of the policy domain and whether or not it is enabled. You can click other tabs to view configured information.

Adding Resources to Policy Domains

NetPoint defines some resource types by default. A Master Access Administrator can define others. After a resource type is defined, both Master Access Administrators and Delegated Access Administrators can add resources of that type to policy domains they administer. When a Delegated Access Administrator is granted administrative rights for a policy domain, that administrator can add resources to the domain.

Using Host Identifiers and Host Contexts

The Master Access Administrator defines host identifiers on the List All Host Identifiers page (Access System Console > Access System Configuration > Host Identifiers link). See "ObPERM cookie" on page 92 for details.

When you add a resource to a policy domain, you select the host identifier for the machine hosting the resource. If the Master Access Administrator has configured host identifiers for machines, you can select the appropriate one from the drop-down list labeled Host Identifiers on the Resource (add) page.

You can use the Host Identifiers feature to create a host context. A host context consists of multiple hosts identified in relation to a single name, a host context name. Instead of adding to a host identifier name the various ways to reference one host, the Master Access Administrator can add corresponding information for multiple hosts to create a context in which all of these hosts share.

A host context is useful if you want to add to a policy domain resources that have the same URL paths on different machines.You want to protect all of these resources in the same way in the same policy domain. In this case, the only variable that distinguishes one set of resources from another is identification of its host machine. Use of a host context provides an efficient way to add the resources for all hosts to the policy domain at once. From the Host Identifiers drop-down lit, you select the host context name. The rest of the information you enter is the same for all of the sets of resources, so you need only specify it once.

You use the Resources tab page to add resources to a policy domain after you create the domain.

To add resources to a policy domain

1. From the Access Manager, click > My Policy Domains, then click the policy domain link.
2. Click Resources.
3. Click the Add button below the message “There Are No Resources Defined,” After you add resources to this domain, they are listed on this page.
4. In the Resource Type field, select an entry.

   NetPoint provides two default resource types, HTTP and EJB. Others may be available if your administrator defined them through the System Console.

   Note: HTTP covers both HTTP and HTTPS resources.

   If host identifiers were created for individual servers, the Host Identifiers field appears.
5. Select the server hosting the resource you want to add.
6. In the URL Prefix list, select a URL string.

   To add a specific resource, enter the remainder of the URL for that resource in the field, to the right of the URL prefix for the resource.
7. In the next field, enter the name of a region to be appended to the URL prefix.

   For example, i f the prefix you selected in the previous step was /your_company, you might enter /sales in this field.

   Note: You need to add the / in front of your entry unless you specified / as the policy root during setup.

   You can later reuse the same prefix but add a different appended region, for example:

   /your_company/marketing

After the newly defined region is saved, it appears in the URL Prefix field.

Note: By default, NetPoint reads URL prefixes and regions as case-insensitive. To change to case-sensitive, the Access Administrator should use the resource matching feature in the Common Information Configuration/Resource Type Definition function within the NetPoint System Console. If you change this setting, you must restart the Access Servers and AccessGates.

8. In the Description field, enter a description of the protected region (whether a policy domain or a policy).

   Completing this field is optional.

9. Determine when you want Access Server caches to be updated:

   - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with information about this new prefix.
   - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.

10. Click Save.

    The Resource page appears again and displays the name of the new resource.

11. Click OK to confirm your change.

12. Repeat these steps to add more resources to this policy domain.

To add additional resources to a policy domain containing resources

1. From the Access Manager, click > My Policy Domains, then click the policy domain link.
2. Select the Resources tab.
3. Click the Add button.

The Resources page appears, as illustrated below.

4. Select a Host Identifier for the resource, if applicable.

   The Host Identifier enables the Access System to distinguish between otherwise identical URL prefixes for resources that might exist on multiple hosts.

5. Select an existing URL prefix to be the basis for the new URL.

   You can see the URL prefixes for existing resources only if you are a Master Access Administrator or if you are a Delegated Access Administrator with rights to view or manage the URL prefix.

6. Enter the URL prefix for the resource using an acceptable format.

   For example:

   - Directory (/marketing/.../)
   - Directory with wildcards (subfolder/*.html)
   - Specific file (marketing/subfolder/marketing.html)

7. Enter a description.

8. Select the Update Cache check box to add this URL prefix immediately to the Access Server cache.

9. Click Save.

10. **Optional**—Add another resource by repeating steps 3 through 9.

Modifying a Resource's Description

Only the Master Access Administrator can modify a resource's description.

You can modify only the Description field of a resource. If you want to change the resource itself, you must delete it and create a new one.

To change a resource description

1. From the Access Manager, select My Policy Domains.
2. In the My Policy Domains page, click a policy domain's link.

   The policy domain's General page appears.
3. Click the Resources tab.

   The Resource page appears with a list of resource types included in this policy domain.
4. Click a resource's link.

   The next page shows the type and prefix of the resource.
5. Click Modify.

   A new page appears.
6. Change the Description as needed.
7. Click Save.

Deleting a Resource

Only the Master Access Administrator can delete resources.

To delete a resource

1. From the Access Manager, click My Policy Domains, and select a policy domain's link.

   The General page displays the Name, Description, and Enabled status of the domain.
2. Click Resources.

   The Resource page appears.
3. Select the check box for the resource you want to delete and click Delete.

   A message asks you to confirm your decision.
4. Select or deselect the Update Cache field.
5. Click OK to delete the prefix (or click Cancel to exit the page without saving).

Configuring the Master Audit Rule

The NetPoint Access System provides the capability to capture and record user activities for protected resources, including user identity information and information about various authentication and authorization activities. Administrators use auditing information to monitor activity for a specific policy domain.

NetPoint provides a Master Audit Rule that can be configured by a Master Access Administrator. Delegated Access Administrators can use the Master Audit Rule to create their own audit rules for policy domains and policies.

The Access System does not log any audit information to the audit log file until the NetPoint Administrator or Master Access Administrator creates a Master Audit Rule.

The Master Audit Rule contains the following information:

- User identity attributes you want to audit (cn, uid, and so forth)
- Events to audit (authentication success, failure, and so forth)
- Selection of a date format
- Format and event mapping for audit log

Note: Making most parameters unchangeable enforces common auditing parameters across all Access Servers.

NetPoint Administrators and Master Access Administrators use the Access System Console to configure a Master Audit Rule, using the Add the Master Audit Rule page.

To configure a server's Master Audit policy

1. From the Access System Console, click Access System Configuration > Common Information Configuration > Master Audit Rule.
2. Click the Add button on the No Master Audit Rule found page to create the master audit rule.

The Add the Master Audit Rule page appears, as illustrated below.

3. In the Profile Attributes field, enter the identity profile attributes you want to capture.

   These attributes are written to the log file when the event happens. In most cases, cn is the best choice.

   Click the plus (+) and minus (–) icons to add or remove attribute fields.

   The Master Access Administrator can add attributes to this field, but cannot delete the ones you select.

4. In the Audit Events field, select the events you want to capture.

   Master Access Administrators and Delegated Access Administrators can add or delete events when configuring policy domains.

5. In the Audit Event Mapping field, enter the strings logged for each event.

   For example, Authentication Success maps to AUTHENT_SUCCESS.

6. In the Audit Date Type field, select the format in which dates are logged.

7. In the Audit Escape Character field, type a character that separates fields and ensures that logged information appears correctly in reports.

   If no escape character is specified, audit records will not be escaped.

8. In the Audit Record Format field, enter data types associated with authentication and authorization activities.

Note: Supported data types for output to a file are shown below. You may want to output to a database (using audit-2-db, for example). In this case, the format string for audit output must be replaced, as described in the auditing information in *Volume 1*.

- **ob_ip**—Corresponds to the IP address of the machine making the request.
- **ob_datetime**—Corresponds to date and time. The date is logged in the format specified in the master audit policy. The time is logged as hh:mm:ss. The time is always the GMT time on the host that received the request, followed by the host's offset from GMT.
- **ob_serverid**—Corresponds to the ID of the Access Server that is auditing this information.
- **ob_url**—Request URL.
- **ob_operation**—HTTP operation, such as GET, PUT, POST, or others.
- **ob_event**—A string corresponding to the event that occurred. The event can be one of the following: Authentication Success, Authentication Failure, Authorization Success, or Authorization Failure.
- **ob_userid**—Contains the user's distinguished name, if the user was successfully authenticated.

  If the user is authenticated and has an entry in the directory, in addition to the distinguished name, the log may contain other information that the authentication module of the Access Server is configured to audit. If the user does not exist in the directory, the only information that can be audited is the user name. If the user exists in the directory but enters an incorrect password, there is no way to confirm the user's identity. As a result, this information is not audited. Passwords are never written to the audit log for users who do not log in as Anonymous.
- **ob_wgid**—ID of the AccessGate that received the request.
- **ob_date**—Corresponds to date only. It does not include the time of the event unless the date format is ISO.
- **ob_time**—Corresponds to the GMT time at which the event occurred on the host. Time is always logged as hh:mm:ss+/-<offset from GMT on host>.
- **ob_time_no_offset**—Corresponds to the GMT time on the AccessGate, but no GMT offset is logged. Time is logged as hh:mm:ss. Master Access Administrators and Delegated Access Administrators cannot change these settings.

- **ob_reason**—Returns information for authentication success, authentication failure, authorization success, and authorization failure events. The overall reason is either ALLOW (for success) or DENY (for failure). However, in the case of DENY, any of the following reasons, which are given by the minor status code, can be the cause for denial of access. Also, a code indicating that there is no reason may be provided when the event is authentication success or authorization success.

  These reasons are returned to clarify the cause of denial, and they are represented by the following integers:

  - **40**—An invalid password was provided as input to the authentication process.
  - **68**—The overall result of evaluation of the authorization expression was inconclusive.
  - **2**—No reason is provided. This code is returned for authentication success and authorization success events.

9. Determine when you want Access Server chaches to be updated.
   - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with this auditing information.
   - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read the new auditing information from the directory server.
10. Click Save to implement your changes (or Cancel to leave this page without saving).

Modifying the Master Audit Rule

NetPoint Administrators and Master Access Administrators use the Access System Console to modify a Master Audit Rule. You use the Modify the Master Audit Rule page to change the configuration of the Master Audit Rule.

To modify the Master Audit Rule

1. From the Access System Console, select Access System Console > Access System Configuration > Common Information Configuration > Master Audit Rule
2. Click the Modify button on the Master Audit Rule page.

   The Modify the Master Audit Rule page appears.
3. Change the parameters as necessary.
4. Click Save.

Deleting the Master Audit Rule

NetPoint Administrators and Master Access Administrators can delete the existing Master Audit Rule from the Master Audit Rule page.

To delete the Master Audit Rule

1. From the Access System Console, select Access System Configuration > Common Information Configuration > Master Audit Rule
2. In the Master Audit Rule page, click Delete.

   You are prompted to confirm your decision.
3. Click OK to delete the rule (or Cancel to exit without saving).

Configuring Policies

Policies enable you to differentiate how subsets of resources in a domain are protected. You can use policies to establish more or less stringent protection for a subgroup of resources of a policy domain.

A policy can include:

- One or more resources.
- Allowed operations (request methods) for a resource type.
- URL patterns for a specific file, directory, query string pattern, or query string variable.

  See “URL Prefixes” on page 112 for details.
- Authentication and auditing rules, and authorization rules and expressions different from the default ones.

If a resource is not covered by a policy, the default rules of the domain apply.

The following example of a policy domain includes two policies. Boggle Games, Inc. provides human resources information to three categories of personnel: regular employees, part-time employees, and contracted employees. The policy domain includes one URL: /mycompany/HR. Other details of the policies are:

- The company shares some information with all groups of users. All users know how and where to obtain a building access badge.

  Badge information resources reside in a subordinate badge directory, /mycompany/HR/badges

  However, because resources in the badges directory are not protected with a policy, they fall under the protection of the policy domain’s default rules.

- The company shares some information only with regular employees; regular employees can view information about holiday, vacation, and stock benefits.

  A policy is used to protect the resources for employee benefits, which reside in directories subordinate to /mycompany/HR.

- The company shares some information only with managers; managers can view lists of preferred vendors who provide contract personnel to Boggle Games, Inc.

Policies with Overlapping Patterns

If you have multiple policies with overlapping patterns, the order of the policies within the policy domain becomes important. In this case, you should order the policies from the most granular to the least granular.

Adding a Policy

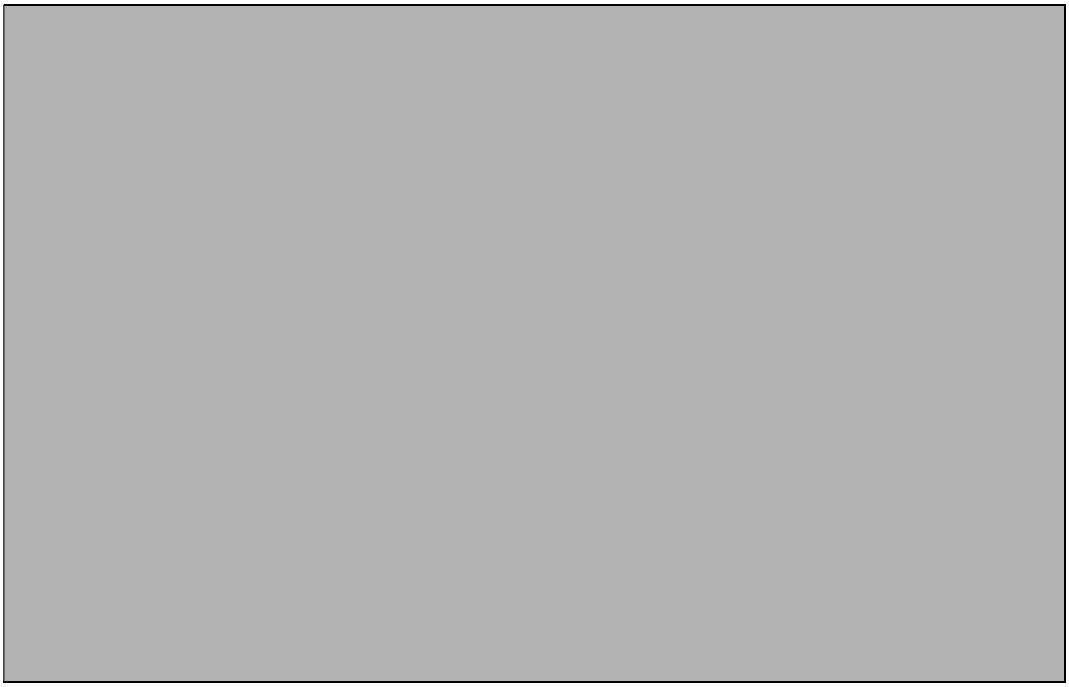
You use the Policies tab page to add a policy to resources of a policy domain.

Note: On some directory servers, adding a very large number of policies and resources may cause a size limit error. In lab conditions, this maximum has only been reached when multiple thousands of resources have been added to the policies.

To add a policy

1. From the Access Manager tab, select My Policy Domains, and select the policy domain that you want to add the policy to.
2. Select the Policies tab and click Add.

   The following page appears.



3. Fill in information for the policy.
4. Click Save.

Modifying a Policy

You use the Policies page to modify a policy.

To modify a policy

1. From the Access Manager tab, select My Policy Domains and click the link for the policy domain whose policy you want to modify.
2. Select the Policies tab, select the policy, and click Modify.
3. On the Policies tab modification page, change the policy information.
4. Click Save.

Setting the Order in which Policies Are Checked

If you create two or more policies, you can specify the order in which the Access Server checks them. By default, a new policy is checked last.

To set the order of policies within a domain

1. From the Access Manager, select My Policy Domains, and click the link for the policy domain.
2. Select the Policies tab.
3. Click Order.
4. Select the name of the policy you want to move within the current order.

   Click the Up and Down arrows to relocate the policy.

   Repeat this process for each of the policies whose order you want to change.
5. Determine when you want Access Server chaches to be updated.
   - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with this auditing information.
   - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read the new auditing information from the directory server.
6. When you are satisfied with the order of the list of policies, click Save.

Deleting a Policy

You delete a policy directly from the list of policies for the policy domain it belongs to.

To delete a policy

1. From the Access Manager tab, select My Policy Domains, click a link for a policy domain, then select the Policies tab.
2. Select the check box before the name of the policy you want to delete.
3. Click Delete.

Deploying a Policy into Production

After you have tested a policy domain that you administer, and you are satisfied that resource protection is enacted as planned, you can deploy the domain for production use. To deploy a policy domain, you enable it.

You must also enable a policy domain to test it. See “Using Access Tester” on page 143 for information describing how to test a policy domain.

Auditing User Activity for a Policy Domain

Auditing is the process of collecting information about users' activities in relation to the resources of a policy domain or its policies. The Access System automatically audits administrative events, such as clearing information from caches. Audit policies set in the Master Audit Rule and audit rules derived from it determine what is tracked.

You can configure audit policies for:

- Authentication and authorization success or failure
- Resource access
- Policy modification

You can customize audit output to include user profile attributes.You can use audit trails for reporting, history, or any purpose you see fit. For example, you can collect the cn and other attributes of user profiles to maintain detailed information about policy domain usage. This information can be searched and used to generate reports.

Creating an Audit Rule for a Policy Domain

You can create audit rules for a policy domain. A policy domain's audit rule serves as the default rule for all resources of the domain unless you define an audit rule for any of the domain's policies.

You must derive this rule from a Master Audit Rule created by a Master Access Administrator. For details about creating a Master Audit Rule, see "Configuring the Master Audit Rule" on page 132.

To create an audit rule for a policy domain

1. From the Access System Console, select Access Manager > My Policy Domains, and click a link for a policy domain.

   The General page for the selected policy domain appears.

2. Select the Default Rules tab.

3. Select Audit Rule.

   A page appears either showing the audit rule defined for the policy domain or reporting that there is no rule defined. If the page states that there is no Master Audit Rule defined, a Master Access Administrator must create one before you can define an audit rule for the policy domain.

4. Click Add to start the audit rule.

   The Audit Rule page appears. If the Master Audit Rule exists, its values are shown as defaults.

5. Select the events to be audited and the audit profile attributes.
6. Click Save.

Modifying an Audit Rule for a Policy Domain

For a policy domain, you can modify existing audit rules, which are derived from the Master Audit Rule.

To modify an audit rule for a policy domain

1. From the Access System Console, select Access Manager > My Policy Domains, and click a link for a policy domain.

   The General page for the selected policy domain appears.
2. Select Default Rules > Audit Rule.

   The General page appears.
3. Click Default Rules > Audit Rule tab.

   The Audit Rule page appears.
4. Select the audit rule to be modified.

   A page with the rule's information appears.
5. Click Modify.

   The rule's page with editable text fields appears.
6. Modify the information and click Save.

Defining an Audit Rule for a Policy

If you define an audit rule for a policy, it overrides the default one defined for the policy domain. Before you can define a policy's audit rule, a Master Access Administrator must create a Master Audit Rule.

To define an audit rule for a policy

1. From the Access System Console, select Access Manager > My Policy Domains, and click a link for a policy domain.

   The General page for the selected policy domain appears.
2. Click the Policies tab.
3. Select the policy for which you want to create an audit rule.

4. Click Audit Rule.

   A page appears either showing the audit rule defined for the policy domain or reporting that there is no rule defined. If the page states that there is no Master Audit Rule, a Master Access Administrator must create one before you can define an audit rule for the policy.

5. Click Add to start an audit rule.

   The Audit Rule page appears. Values for the Master Audit Rule, if one exists, are shown as defaults.

6. Select the events to be audited and the audit profile attributes.

7. Click Save.

Modifying an Audit Rule for a Policy

You can modify the audit rules for the policies of a policy domain. These rules are derived from the Master Audit Rule created by a Master Access Administrator.

To modify an audit rule for a policy

1. From the Access System Console, select Access Manager > My Policy Domains, and click a link for a policy domain.

   The General page for the selected policy domain appears.

2. Click the Policies tab.

3. Select the policy for which you want to create an audit rule.

4. Click Audit Rule.

   The Audit Rule page appears.

5. Select one of the audit rules.

   A page with the rule's information appears.

6. Click Modify.

   A page with the rule's information in editable text fields appears.

7. Modify the information, and click Save.

About the Audit Log File

An audit rule causes event-based data to be written to the audit log file. There is one audit log per Access Server. You can configure the size of the audit log file and the rotation interval per server. Depending on events recorded, the audit log may contain some duplicate audit entries.

Note: To audit to a database, by using audit-2-db for example, the format string used for audit output must be replaced, as documented in *Volume 1*. Also, you need to have a supported database installed and specific configuration in NetPoint.

Using Access Tester

Use the Access Tester to verify that the authentication and authorization rules and authorization expressions you created for a policy domain produce the results you expect. You should test the policy domain before you make it available for production. After you select various parameters for your rules and compare the results to what you expect, you may need to make adjustments to your rules.

Note: You must enable the policy domain before you can test it. See "Enabling and Disabling Policy Domains" on page 125 for details.

To run Access Tester

1. From the Access Manager, click the Access Tester link in the left navigation bar.
2. In the URL field of the Access Tester page, type the full path to the application or content you want to check.
3. In the Resource Type field, select an entry from the list.

   You can only select a type that has been defined in the NetPoint System Console.
4. In the Resource Operation field, select the request methods you want to test for this URL.

   Note: The operations available in this field depend on the resource type you selected in the previous step.

   If you select none of these, they *all* are tested.
5. If you want to know if a particular computer can access the resource (URL), type the computer's IP address into the From this IP Address field.

   You must enter a complete IP address. Wildcards are not allowed in this field.

6. In the Date/Time of access list, do one of the following:
   - Click the button beside “any” to test this resource without timing restrictions.
   - Click the button beside “specific date and time” and fill in the following Date and Time fields.
7. In the Check access for the following user(s) field, do one of the following:
   - Click the button beside “all users.”

     If you select all users, the Access Tester processes the authentication and authorization information for all users. If there are a great many user entries in your database, this could take a considerable amount of time.
   - Click the button beside “selected users,” then click the Select User button to display the Sector page where you can select specific users.

     Note: Do *not* select groups. Access Tester can only test access control for individual users, not groups. Also, it will not resolve groups to the individual level.
8. Complete the Access Tester page by clicking the arrows beside Display Options to show available options.

   The options appear.
9. In the Show Administrators field, select the number of end users you want to display at one time.
10. From the list, select the button beside the option that describes the appropriate user access:
    - show only users who are allowed
    - show only users who are
    - show both.

    Policy and Rule options are also listed:(
    - show matching Policy
    - show matching Rule.
11. If this resource (URL) is protected by a policy, and you want it to display the policy, select the button beside show matching Policy.
12. If you want the authorization rule for this resource (URL) to be displayed, select the button beside show matching Rule.
13. Click Submit.

    The results appear.

Delegating Policy Domain Administration

When a Master Access Administrators creates a policy domain, he or she assumes the role of default Delegated Access Administrator. This default Delegated Access Administrator has all management rights within that domain, and can delegate administration of that domain to others who then become Delegated Access Administrators.

There are three levels of rights for Delegated Access Administrators:

- **Delegate**—Delegates grant or basic rights to other users.
- **Grant**—Delegates basic rights to other users.
- **Basic**—Performs delegated tasks, but cannot delegate this right to others.

Only Delegated Access Administrators who have rights to a specific domain—or the Master Access Administrator—can view a policy domain.

Delegated Access Administrators can manage policy domains that are delegated to them. They can also *create* policy domains for resources that fall under the URL prefixes of the policy domains that are delegated to them.

Table 6 summarizes the rights of the different types of administrator:

Table 6 Types of administrators and their rights.

Type of Administrator	Policy Domain Rights
NetPoint Administrator	• Creates Master Access Administrators. • Creates the policy root. • Creates the policy base.
Master Access Administrator	• Creates the first policy domain and adds resources to it. • Defines resource types. • Creates, deletes, and manages authentication and authorization schemes. • Creates the Master Audit Rule. • Delegates management of policy domains to Delegated Access Administrators. • Retains all rights delegated to other users.
Delegated Access Administrator with delegate rights	For that policy domain only, a Delegated Access Administrator with delegate rights can: • View the domain. • Create authentication and authorization rules. • Create an authorization expression for the policy domain and for any policies it contains. • Create audit rules based on the Master Audit Rule. • Define Delegated Access Administrators with grant or basic rights. • Enable or disable the policy domain. • Test the policy domain. **Important:** The Delegated Access Administrator cannot redefine the attributes of the Master Audit Rule.

Type of Administrator	Policy Domain Rights
Delegated Access Administrator with grant rights	Created by a Master Access Administrator or a Delegated Access Administrator with delegate rights. For that policy domain only, a Delegated Access Administrator with grant rights can: • View the domain. • Create and delete authorization rules. • Create or modify an authorization expression for the policy domain and for any policies it contains. • Create audit rules based on the Master Audit Rule, or change the events to be audited, removing existing events or including other ones. • Create and delete policies for resources in the policy domain. • Define Delegated Access Administrators with grant or basic rights. • Enable or disable the policy domain. • Test the policy domain.
Delegated Access Administrator with basic rights	Created by a Master Access Administrator or a Delegated Access Administrator with delegate or grant rights. A Delegated Access Administrator with basic rights cannot create or delete policy domains. For the specified policy domain, this administrator can: • View the domain. • Create or delete authentication and authorization rules. • Create or modify an authorization expression for the policy domain or any of its policies. • Create audit rules based on the Master Audit Rule. • Redefine the events to be audited, removing existing events or including other events. • Add new attributes to the Master Audit Rule. However, this administrator cannot redefine existing attributes. • Create and delete policies for resources. • Enable or disable the domain. • Using Access Tester, verify access to the resources protected by the policy domain.

Configuring Policy Domain Administrators

Both Master Access Administrators and Delegated Access Administrators can administer policy domains. For details about creating Master Access Administrators, see “Configuring Access Administrators” on page 22. To create and view Delegated Access Administrators for a policy domain and to modify delegated rights, see the following paragraphs.

To view Delegated Access Administrators for a policy domain

1. From the Access Manager, select My Policy Domains and click the policy domain.
2. Select Delegated Access Admins.
3. On the Delegated Access Admins page, in the Show Administrators with field, select the Delegate Rights, Grant Rights, or Basic Rights radio button.

   The page is refreshed to display the current users and groups with the selected administrative right for this policy domain. If no users have this right, you receive the message “There are no Delegated Access Admins with this right.”
4. Click the administrator link to display the profile for the user or group.

To delegate rights for a policy domain

1. From the Access Manager, select My Policy Domains and click the policy domain.
2. Select Delegated Access Admins.
3. Click the radio button for the kind of right that you want to grant.
4. Click the Modify button at the bottom of the Delegated Access Admin page.
5. Click Select User.
6. Use the Search process to display a list of users to select from, and click Done.
7. Click Save.

To modify policy domain rights

1. From the Access Manager, select My Policy Domains.
2. Click the policy domain.
3. Select the Delegated Access Admins tab.
4. Click Modify.
5. Modify the field values for the rights you want to change.
6. Click Save.

4 Configuring User Authentication

The NetPoint Access System enables you to protect your resources with policy domains, which contain rules that determine who can access them. Policy domains include authentication rules. Authentication is the process of proving that a user is who he or she claims to be. For the NetPoint Access System, how authentication of users is to be performed is specified by the content of authentication schemes, which are included in authentication rules. Policy domains can include policies, which are used for specific resources to define finer-grained protection for those resources. Policies can also contain authentication rules.

Policy domains and policies also include authorization rules and expressions, and audit rules, which are described in other chapters of this guide. After you have created your policy domains by identifying their resources, you can define their schemes, rules, and expressions. You can create the authentication rules, authorization rules and expressions, and audit rules for a policy domain in any order.

This chapter explains authentication schemes and authentication rules. It also explains actions, which you can associate with the possible outcomes for authentication rules. The chapter explains how to create, use, and manage these schemes, rules, and actions.

This chapter contains the following topics:

About Authentication

You can use the NetPoint Access System to define authentication schemes and authentication rules to establish ways in which to authenticate users requesting access to the resources of your policy domains. To authenticate a user, you obtain and process information about the user to verify that the user is who he or she claims to be.

Background Reading

Before you read this chapter, read the following:

- “Configuring AccessGates and Access Servers” on page 33.

  This chapter describes how to configure AccessGates and Access Servers, which you must do before the policy domains you create can take effect.

- “Protecting Resources with Policy Domains” on page 95.

  This chapter describes policy domains, policies, resources, and the Master Audit Rule.

- “Configuring User Authorization” on page 229.

  This chapter describes creating authorization schemes, rules, and expressions.

Authentication Basics

To configure authentication, you create the following components:

- **Authentication Schemes**—An authentication scheme includes the method used to challenge the user for credentials. It also includes one or more steps consisting of one or more plug-ins used to perform different parts of the authentication process.

- **Authentication Plug-Ins**—NetPoint provides default plug-ins that implement certain methods used to challenge the user for credentials. Netpoint also provides a credential mapping plug-in to map credentials obtained from a user requesting access to a resource to a user profile in the NetPoint LDAP directory. You can use these plug-ins alone, you can replace them with custom ones, or you can use them in combination with custom ones.
- **Authentication Rules**—Authentication rules include authentication schemes. For each policy domain, you provide one default authentication rule. You can also create one authentication rule for each of a policy domain's policies.

You can use the NetPoint Access System to obtain user information to authenticate users under the following conditions:

You can use the NetPoint Access System to obtain user information to authenticate users under the following conditions:

- If you store all of your user information in one branch of a single directory.
- If you store all of your user information in more than one directory (using the same schema)
- If you have divided storage of your user profile information logically across different branches of your directory, each with its own search base.

Searching a Single Directory—You can use authentication to search a single location (a single search base of a single directory). For example, an organization may want to limit to a single directory the search for user information required for authentication. If the information is not found in that directory, the user cannot be authenticated and the search terminates.

Searching Two Directories of the Same Type—You can use chained authentication to search two or more directories of the same type managed by the same COREid system.

- **Searching Two Directories Consecutively**—An organization may use two directories of the same type to store information about its employees. The organization may want each directory to be searched until information about a user is found. If the information is found in the first directory, the organization may want to terminate the search process. If the information is not found in the first directory, the organization may want to continue the process and search the next directory. Alternatively, the organization may want to end the search if the user information is not found in the first directory, depending on the user's status.
- **Searching One Directory or Another Based on Conditions**—The same organization may want to create another chained authentication scheme used to search one or another directory. The scheme may specify that one directory is to be searched if the user is an employee and that another directory is to be searched if the user is a vendor. For each condition, if the user information is

not found in the first directory, the scheme specifies that a third directory is to be searched before the authentication process is terminated.

Searching Different Branches of the Same Directory—You can use chained authentication to search different branches of the same directory for user profile information. An organization may store some user profiles in one branch, some in another, and some in yet another. The third branch of the directory may contain legacy data. The organization may search the third branch for a user profile only if the information for the user cannot be found in the other two branches. For this purpose, the organization can configure an authentication scheme whose steps contain plug-ins to begin from the first search base, map the user's credentials to a user profile, and, if it is found, process the credentials, then terminate the search. If the user profile is not found in the first branch, the scheme's steps can direct the search to the next search base, and so on, until the user profile is found, or not.

Authentication Schemes

An *authentication scheme* specifies how authentication is to be performed for users requesting access to a resource protected by the authentication rule that contains the scheme. A simple authentication scheme can contain a single step. For *chained authentication*, an authentication scheme contains multiple steps linked together to produce different behaviors depending on certain conditions.

Authentication schemes include four main components:

- **General Information**—To describe an authentication scheme, you configure its general information. This information includes data such as the method to be used to challenge the user for credentials authenticating his or her identity and the security level the scheme provides. For details, see "Creating an Authentication Scheme" on page 154.
- **Plug-Ins**—The plug-ins you add to an authentication scheme are fundamental to it. You can use plug-ins provided by NetPoint and custom plug-ins. Only the plug-ins you add to a scheme can be used for any of its steps. For details, see "Plug-Ins for Authentication" on page 165.
- **Steps**—An authentication scheme can include one or more steps, each of which must include at least one plug-in. A *step* provides a way to create a discrete group of plug-ins executed in order of their position in the step. To connect the steps of a chained authentication scheme, you specify the step to be executed next, depending on the outcome of the present step. A different step may be executed next if the present step fails or if it succeeds. You can repeat a step in an authentication scheme. You can stop the authentication process after a step. For details, see "Authentication Steps" on page 186.
- **Authentication Flows**—Authentication flows are the possible execution paths through the steps of an authentication scheme. For a single-step scheme, the

authentication flow consists of execution of the plug-ins of the step. For details, see “Authentication Flows” on page 197.

Creating Authentication Schemes

Every authentication scheme must contain a challenge method and a way to map the credentials provided by the user to the corresponding user profile stored in the directory. Creating an authentication scheme includes defining how the scheme challenges the user for credentials, maps the information, verifies it, and so forth. For example, a scheme’s challenge method may require users to provide passwords or it may require users to provide certificates attesting to their identity.

An authentication scheme can also contain plug-ins that do additional processing, such as search multiple directories based on conditions and perform tasks based on the outcome of other processes.After you create an authentication scheme, you can add plug-ins to the scheme and then configure the scheme’s steps and their execution order.

Only Master Access Administrators can create authentication schemes. See “Delegating Policy Domain Administration” on page 145.

Task overview: Create an authentication scheme

1. Provide general information about the scheme to define it, which includes specifying the scheme’s challenge method (General page). See “Creating an Authentication Scheme” on page 154.
2. Add to the scheme the plug-ins and their parameters to be used for any of the scheme’s steps (Plugin page). See “Adding Plug-Ins and Managing Them” on page 178.

   Among the plug-ins you add are ones to perform required tasks—such as mapping a user’s credentials to a user profile—and optional ones to perform tasks specific to your environment. You can select from among the plug-ins provided by NetPoint and any custom ones you have created.
3. Create one or more steps for the scheme, and name each step (Steps page). See “Configuring and Managing Steps” on page 191.
4. Add plug-ins to the named step (Steps page).

   You add plug-ins to a named step when you create the step. You select plug-ins for a step from among those you added to the scheme.
5. Define the authentication flows—the flows of control through the scheme’s steps (Authentication Flows page). See “Authentication Flows” on page 197.
6. Test the authentication flow and verify it to ensure it does not contain any loops called cycles, which could cause endless, repeated execution of the same plug-ins (Authentication Flows page).

7. If an authentication flow contains cycles, correct the flow (Authentication Flow page).

Listing Authentication Schemes

Before you create an authentication scheme, list the existing ones to ensure that the one you want to create is not already defined. When you list authentication schemes, the list shows any new authentication schemes and any authentication schemes created for versions of NetPoint prior to NetPoint 6.5. Pre-existing schemes are converted to authentication schemes containing a single default step.

Note: Once you modify a pre-existing scheme, it cannot be used for systems prior to NetPoint 6.5.

To view a list of authentication schemes

1. From the NetPoint Access System Console, click Access System Configuration > Authentication Management.
2. View the Authentication Management: List All Authentication Schemes page that appears.

   This page displays a message stating there are no authentication schemes configured, if that is the case.

Creating an Authentication Scheme

An authentication scheme is defined and identified by information you specify using the General tab's Define an Authentication Scheme page. Before you define an authentication scheme, you need to determine the following:

- A name for the scheme and a brief description of what it does.

  An authentication scheme must have a name that is unique among all authentication schemes you create. Delegated Access Administrators who create authentication rules containing the scheme, will select a scheme from among existing schemes. Providing a brief description of each scheme makes it easier for them to do so.

- The security level of the authentication scheme.

  The security level of the scheme reflects the kind of challenge method and degree of security used to protect transport of credentials from the user. The security level is expressed as an integer.

  The security level of a scheme also affects the single sign-on user capability. After an end user is authenticated for a resource at a specified level, the user is automatically authenticated for other resources within the same policy domain or in different policy domains, *if* the resources have the *same or a lower security level* as the original resource. For details about how to change the

level, see “Changing the Security Level of an Authentication Scheme” on page 168.

- The type of challenge and its parameters to be used to obtain the user’s credentials

  The *challenge method* specifies how authentication is to be performed and the information required to authenticate the user. Each authentication scheme can have only one challenge method. Authentication is successful if the user credentials obtained in response to a challenge match only one DN in the directory—not more than one or none.

  Usually a challenge parameter provides WebGate with additional information to perform an authentication, often used to prompt the user for information. Challenge parameters are entered in *name*:*value* format.

- Whether users must be authenticated using a server enabled for Secure Sockets Layer (SSL).

  For information about single sign-on, see “Configuring Single Sign-On” on page 295.

- The URL of a server specified as the Challenge Redirect, if you want user requests to be redirected to another server for processing.

  Authentication schemes may require redirection of the request to another URL to properly carry out the authentication. For example, redirection is used when an authentication request for a resource is made over HTTP but the authentication scheme requires the authentication to be made over HTTPS (secure HTTP). WebGate sends the redirect to the user’s browser telling it to request a URL defined by the authentication scheme. After authentication is completed, WebGate redirects the browser back to the original requested resource.

  Also, redirection is required to perform multi-domain single sign-on (SSO). For information describing how challenge redirects are used for multi-domain single sign-on, see “Multi-Domain Single Sign-On” on page 304.

- Whether the scheme should be enabled.

  This page includes a radio button that you can set to enable or an authentication scheme. For details about enabling and disabling a scheme, see “Enabling and Disabling Authentication Schemes” on page 164.

- Whether the Access Server’s cache should be updated automatically with new information and changes you make to the scheme

  This page includes a checkbox that you can select to specify that the cache should be updated.

To create an authentication scheme

1. From the Access System Console, click Access System Configuration > Authentication Management
2. Click Add on the Authentication Management: List All Authentication Schemes page.

   The Define a new authentication scheme page of the General tab page appears, as illustrated below.

3. In the Name field, specify a name for the authentication scheme.

   Each authentication scheme must have a unique name.
4. In the Description field, provide a brief description of the scheme.

   For instance, you might explain the purpose of the scheme and its behavior.
5. In the Level field, enter an integer corresponding to the level of security of the scheme.
6. In the Challenge Method field, click the radio button for the authentication scheme challenge method you want to use (each authentication scheme can have only *one* challenge method see “About Challenge Methods” on page 158):
   - None
   - Basic
   - X.509
   - Form
   - Ext

7. If you selected Form, Basic, or Ext for the challenge method, specify a Challenge Parameter.

   - For the basic Challenge Method, type a short text string to be used as a hint to help end users remember their usernames and passwords for the requested resource.

     Here is an example of the text string for an LDAP directory:

     `realm:LDAP username + password`

   - If you selected the Form challenge method, you are required to provide the following three parameters in the Challenge Parameter fields.

Challenge Parameter	Description
`form:`	Indicates where the HTML form is located relative to the host's document directory. For example, `form:/login.html`.
`creds`:	Lists all fields used for login in the HTML form. The parameter creds is a space-separated list. For example: `creds:login password`. Note: You can specify the creds parameter for the other types of challenge methods.
`action`:	The URL that the HTML form is posting to.

A fourth parameter—`passthrough`—is optional.

`passthrough`:	This parameter value determines whether the WebGate redirects the browser back to the original requested resource or passes the login credentials on to another program. NetPoint assumes that the URL given for the form in the authentication scheme is on the same machine as WebGate. Possible values are yes or no: • Accept the default value of no if you want WebGate to redirect the browser back to the original requester resource. • Specify yes if you want to pass the login credentials through to a post-processing program.

8. Determine whether you want the end user authenticated through an SSL-enabled server.

   If you click Yes, the request is routed to the HTTPS server you specify in the Challenge Redirect field.

9. In the Challenge Redirect field, enter the URL of another server to which you want to redirect this request if authentication does not take place on the original server.

   Use the host URL of the designated primary authentication server. For example:

   https://www.*yourcompany*.com

10. Select the radio button to enable or disable the authentication scheme.

11. Click Save (or Cancel):

    - If you click Save, the Details for an Authentication Scheme display page appears. This page displays the information you entered for the new authentication scheme.
    - If you click Cancel, the configuration is not saved and the page listing all authentication schemes is displayed again.

About Challenge Methods

You must include a challenge method in every authentication scheme you define. For your authentication schemes, you can use a predefined challenge method, provided by NetPoint, or a custom one.

NetPoint supports the following five challenge methods:

- **None**—Users are not prompted to provide any credential information. This method allows access to NetPoint-specific resources (URLs) you do not want protected with the Access System, for example, Self Registration.
- **Basic**—Users must enter a username and password in a pop-up window supplied by the Web server. This method can be redirected to SSL. For additional information, see “Basic and Client Certificates” on page 159.
- **Client Cert (X509Cert)**—X.509 digital certificates over SSL. A user’s browser must supply a certificate. For additional information, see “Basic and Client Certificates” on page 159.
- **Form**—This method is similar to the basic challenge method, but users enter information in the custom HTML form. You can choose the information users must provide in the form that you create. For details about form-based authentication for redirecting users to another site, see “Form-Based Authentication” on page 349.

- **Ext**—An external challenge method (outside NetPoint) is used. Allows you to use your own authentication challenge method.

  If you use Ext, you must provide the challenge parameter: creds. This parameter is a space-separated list of server variables set by the external challenge method. See the *NetPoint 7.0 Customization Guide* for more information.

Basic and Client Certificates

NetPoint supports client certificate authentication using public key encryption cryptography and X.509 certificates. The client certificate challenge method uses the Secure Sockets Layer version 3 (SSLv3) certificate authentication protocol built into browsers and Web servers. Authenticating users with a client certificate requires the client to establish an SSL connection with a Web server that has been configured to process client certificates.

For both Basic and X.509, you can configure an AccessGate to handle unauthenticated requests received over a non-SSL connection.

For the client certificate challenge method, you configure the AccessGate to redirect the user's browser to another server to establish an SSL connection, as mentioned previously. After the AccessGate authenticates the certificate, it redirects the user's browser back to the original URL.

Schemes and Policy Domains Configured During Installation

If the NetPoint Administrator selected a challenge method during installation of the Access System, NetPoint configures authentication schemes automatically. The following authentication schemes provided by NetPoint include a single step.

- **Basic**—The user must type the username and password in a pop-up window supplied by the server.

  The username and password are verified against the user's User Profile in the LDAP directory.

  Note: If you are using the Oblix-provided schemes, you must be sure the obMappingFilter of the plug-in parameter is set correctly for your directory and environment. For details, see Table 8, "Credential Mapping Parameters," on page 171.

- **Client Certificate**—The user must supply a digital certificate to the policy domain to complete authentication.

  NetPoint supports X.509 certificates. The user's organization can determine how to obtain a certificate; there are no NetPoint requirements in this regard.

- **NetPoint None Authentication**—This method is used to unprotect specific NetPoint URLs. Users are not prompted to provide any credential information. This method allows access to NetPoint-specific resources (URLs) that you do not want protected with the Access System, for example, Self Registration and Lost Password Management.

  This authentication scheme maps the credential_mapping to Anonymous User.

- **NetPoint Basic over LDAP**—Protects NetPoint-related resources (URLs).

- **NetPoint Basic over LDAP for AD Forest**—Protects NetPoint-related resources (URLs) for AD Forest.

See the *NetPoint 7.0 Installation Guide* for information about configuration of these schemes during the installation process.

Modifying an Authentication Scheme

You can modify the content of an existing authentication scheme. Also, as you create an authentication scheme, you can modify any part of it.

Before you modify a scheme, ensure that it is not included in the authentication rules of any active policy domains, and disable the scheme if it is enabled. For details, see “Enabling and Disabling Authentication Schemes” on page 164.

To modify a new authentication scheme as you define it, select the tab and modify the information on its pages. The following procedure describes how to modify an existing scheme.

Note: Existing authentication schemes are compatible with prior releases of NetPoint. However, if you modify an older authentication scheme, it will run on NetPoint 6.5 and later Access Servers but not on earlier versions of the Access Server.

To modify the content of an authentication scheme

1. Ensure that the scheme is not included in the authentication rules of any active policy domains, and disable the scheme if it is enabled.

   For details, see “Enabling and Disabling Authentication Schemes” on page 164.

2. From the Access System Console, click the Access System Configuration tab.

   Access System Console > Access System Configuration

3. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.

4. Click the name of the authentication scheme you want to modify.

   The Details for an Authentication Scheme page appears. From this page, you can select other tabs, such as Plugins, Steps, Authentication Flow.

5. Click Modify.

   The Modifying Authentication Scheme page appears, as illustrated below. You can modify the scheme's general information from this page.

   To modify other parts of the scheme

   - Select the tab for that part.
   - Click Modify on the page which appears.
   - Follow the configuration process for that page.

6. Click Save.

7. Re-enable the scheme.

   For details, see "Enabling and Disabling Authentication Schemes" on page 164. Authentication schemes must be enabled to be available for use in a rule. If a disabled scheme is used in action domains or policies, the resource is not protected.

Viewing an Authentication Scheme Configuration

After you create authentication schemes, you can view their contents.

To view the configuration for an authentication scheme

1. Launch the NetPoint Access System, select Access System Console, then Access System Configuration.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme you want to see.

   The Details for an Authentication Scheme page appears.

Deleting a Authentication Scheme

Before you attempt to delete an authentication scheme, disable it. For details and other requirements, see “Enabling and Disabling Authentication Schemes” on page 164.

To delete an authentication scheme

1. Launch the NetPoint Access System, select Access System Console, then Access System Configuration.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.
3. Select the check box for the authentication scheme that you want to delete.

   To delete more than one scheme, select the check box for each scheme.
4. Click Delete.

Configuring an Authentication Scheme when Using Disjoint Domains

If you have disjoint domains, you need to configure an authentication scheme that enables searches for users with identical user IDs who reside in disjoint domains.

To configure an authentication scheme for disjoint domains

1. On Active Directory, add the plug-in for NetPoint Basic over AD Forest to your authentication scheme.

   See “Adding a Plug-In to an Authentication Scheme” on page 180 for details.

   For other platforms, create a custom authentication scheme similar to the following:

```
credential_mapping
obMappingBase="%domain%",obMappingFilter="(&(&(objectclass=gensiteorgperson)
(genuserid=%userid%))(|(!(obuseraccountcontrol=*))
(obuseraccountcontrol=ACTIVATED)))",obdomain="domain"
```

2. Modify this plug-in.

   Change the object class to your user object class.

   Change the genuserid to your login attribute configured on your user object class.

3. In the authentication action that you define upon successful authentication using this scheme, you need to set the following values

   Type—HEADERVAR

   Name—HTTP_OBLIX_UID

   Return Attribute—obuniqueid

   See “Setting Authentication Actions” on page 216 for details.

   Note: This must be done for both the default identity and access policy domains.

4. In addition, you need to make the following configuration file changes:

   In the following file:

   AccessManager_install_dir/access/oblix/apps/common/bin/globalparams.lst

   change the value of whichAttrIsLogin to ObUniqueID

   Make the same change in the following file:

   COREid_install_dir/identity/oblix/apps/common/bin/globalparams.xml

Managing Authentication Schemes

You may want to modify the definition of an authentication scheme, for example, to change the security level or the challenge RedirectURL if SSL is required. Before you modify information for an authentication scheme, you must disable the scheme. This section describes how to enable and disable authentication schemes, and how to modify them.

Enabling and Disabling Authentication Schemes

The Define an Authentication Scheme page of the General tab includes a radio button which you can set to enable or disable an authentication scheme.

When you create an authentication scheme, the scheme is disabled until you enable it. It is good practice to enable an authentication scheme only *after* you complete its configuration.

To modify any part of an authentication scheme, you must first disable the scheme. Before you disable a scheme to modify it, you should ensure that the scheme is not used in authentication rules of any active policy domains.

If a scheme is disabled:

- It is not available for use in authentication rules.
- Resources previously protected by the scheme are no longer available to users requesting access to them.

The following error message is reported when an attempt is made to access resources protected by an authentication rule containing a disabled authentication scheme:

```
The authentication scheme SchemeID is invalid or has been
disabled
```

After you modify an authentication scheme and enable the scheme, Delegated Access Administrators can use it again in authentication rules for their policy domains or policies.

To enable or disable an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.

3. In the List All Authentication Schemes page, click the scheme you want to enable or disable.

   The Details for an Authentication Scheme page appears.

4. Click Modify

   The Modifying Authentication Scheme page appears, as illustrated below

5. Select the radio button to enable or disable the authentication scheme.

   You must disable a scheme before you can modify it.

6. Click Save.

Plug-Ins for Authentication

An authentication *plug-in* is an executable shared library which participates in the user authentication process. Plug-ins are the engines of an authentication scheme. They implement challenge methods, map user credentials to user profile entries in a directory, process user credentials, perform custom tasks related to the authentication process, and so on.

The steps of an authentication scheme include one or more plug-ins. Before you can add plug-ins to a step, you must add them to the authentication scheme. You must add to the authentication scheme all of the plug-ins to be used for any of its steps.

Authentication schemes contain the following two types of plug-ins:

- NetPoint-provided plug-ins
- Custom plug-ins

About NetPoint-Provided Plug-Ins

NetPoint provides plug-ins to implement the challenge methods it supports by default. These plug-ins include a credential mapping plug-in. Every authentication scheme must include a credential mapping plug-in that maps user credentials to a user profile in the directory. You can use the NetPoint-provided plug-in for this purpose, or you can replace it with a custom one that implements the same behavior. See “NetPoint Plug-Ins for Authentication Challenge Methods” on page 168 for details about these plug-ins and their parameters.

You include plug-ins in a step. If execution of a plug-in provided by NetPoint fails, the step that contains the plug-in fails. For details about steps and plug-ins, see “Authentication Steps” on page 186.

About Custom Plug-Ins

In addition to replacing NetPoint-provided plug-ins with custom ones, you can create custom plug-ins to serve other purposes related to your authentication process. If you use more than one directory to store user profile information, you can create custom plug-ins to be used to search each directory. Also, if you store user profile information for one department in one branch of a directory and user profile information for another department in another branch of the same directory, you may want to search the branches consecutively depending on certain conditions. You can create custom plug-ins for this purpose.

If execution of a custom plug-in fails, the outcome depends on the step to be executed next as determined by the authentication flow of the authentication scheme and the return code returned by the plug-in.

For information describing how to create plug-ins to be used for authentication, see the chapter on the authentication plug-in API in the *NetPoint 7.0 Developer Guide*.

For information about authentication flows, see “Authentication Flows” on page 197.

Return Codes for Plug-Ins

If you create a custom plug-in, the NetPoint Access Server expects your custom plug-in to return one of the following four status codes:

- ObAnPluginStatusContinue
- ObAnPluginStatusAllowed
- ObAnPluginStatusDenied
- ObAnPluginStatusAbort

For details explaining what these return codes means and how the Access Server interprets them and responds to them, see the chapter on the Authentication plug-in API in the *NetPoint 7.0 Developer Guide*.

About Reuse of Plug-Ins

When you add a plug-in to an authentication scheme, the Access Server transparently assigns that plug-in an identifier. The NetPoint Access System manages these numbers internally. You cannot change them or delete them.

Because the Access System uses identifiers to keep track of plug-ins, the execution order of plug-ins is not dependent on their position exclusively, and a single plug-in can be reused in the following ways:

- It can be used in combination with other plug-ins to form a step.
- It can be used more than once within a step. A step can contain multiple instances of the plug-in with different parameters.
- It can be used for different steps of the same authentication scheme.

Reusing Plug-Ins across Authentication Schemes

You can use the plug-ins you create for any number of authentication schemes, but for each authentication scheme, you must rename the plug-in so that its name is unique across authentication schemes.

Changing the Security Level of an Authentication Scheme

You can write a custom plug-in to change the security level of an authentication scheme. In some cases, you may want to increase the security level of an authentication scheme depending on certain conditions. You may want the security level of an authentication scheme to depend on the application the user logged in from. For example, if Active Directory and a reverse proxy are among the sources your users can log in from, you may want to set one authentication security level to be used for users who log in from Active Directory and another security level to be used for users who log in from the reverse proxy.

Your code could determine the source from which the user logged in, and it could set the authentication scheme security level accordingly. It could check the current value of the ObAuthentSchemeLevel variable maintained by the Access Server in the credential list for the scheme. Your plug-in could change the security level, setting the variable value to a security level that depends on the requirements you have established for login from the application. To set the security level, you modify the value of ObAuthentSchemeLevel variable. If you do not change this value, the Access Server uses the security level already set for the authentication scheme through the user interface.

You can use the following code in your plug-in to open the credentials list file, check the ObAuthentSchemeLevel variable value, and set it to the security level you want to use for an application.

```
schemeLevel = pFnBlock->GetCredFn(pInfo->Creds,
“ObAuthentSchemeLevel”);

if (schemeLevel ! = NULL) {
schemeLevelAsInt = atoi(schemeLevel);
schemeLevelAsInt +=10
iota(schemeLevelAsInt, buff, 10);

pFnBlock->SetCredFn(pInfo->Creds,
“ObAuthentSchemeLevel”, buff);
```

NetPoint Plug-Ins for Authentication Challenge Methods

Table 7 shows the predefined challenge methods and the plug-ins that support them. For each challenge method that contains more than one plug-in, the order in which the plug-ins are executed is identified.

You can use these plug-ins in their defined order (as shown in the table) within one or more steps of your authentication scheme; you can use any of them with other plug-ins of your own that provide the required functionality of the plug-ins they replace; or you can provide all of your own custom plug-ins to implement the required ones for the authentication schemes.

Table 7 NetPoint Predefined Challenge Methods and Plug-Ins

Challenge Method	Plug-Ins and Order of Execution
None	1. credential_mapping
Basic	1. credential_mapping 2. validate_password
Client Certificate	1. cert_decode 2. credential_mapping The following plug-ins are optional: 4. selection_filter 5. authn_valicert
Form	1. credential_mapping 2. validate_password

Note: Oblix recommends that all authentication schemes use the credential_mapping plug-in even if you select None as the challenge method. However, this is not a requirement. See “Credential Mapping Plug-In” on page 170 for required parameters.

Here is a description of the plug-ins provided by NetPoint in support of the challenge methods it defines.

- **credential_mapping**—This plug-in maps the user’s userID to a valid distinguished name (DN) in the directory. You can configure the attribute to which the userID is mapped. The most common attribute it is mapped to is uid. However, it is possible for a customer to map the userID to a profile attribute other than uid by changing the obMappingFilter parameter.

  A credential mapping plug-in is required for every authentication scheme. You can use the credential_mapping plug-in provided by NetPoint for an LDAP directory server for this purpose, or you can provide your own plug-in. See “Credential Mapping Plug-In” on page 170.

- **validate_password**—This plug-in is used to validate the user’s password against the LDAP data source. It addresses the Form and Basic challenge methods. See “Validate Password Plug-In” on page 172 for details.

- **selection_filter**—This plug-in further validates the authentication credentials with some criteria. It addresses credentials provided by the user and does not use backend data sources. It addresses all of the challenge methods.

- **cert_decode**—The plug-in validates the certificate and does not use a data source. It addresses the Client Certificate (Cert) challenge method. See “Certificate Decode Plug-In” on page 173 for details.
- **authn_valicert**—This plug-in performs a Certificate Revocation List check for the client certificate. This challenge method plug-in is the only predefined one that is not built into the system. Rather, it is included with the Access Server in a separate library. See “ValiCert Plug-In” on page 177 for details.
- **NT/Win2000**—This plug-in addresses Form and Basic challenge methods for Microsoft Windows 2000 systems. See “Windows NT/2000 Plug-In” on page 227 for details.
- **SecurID**—This plug-in addresses the Form challenge method for SecurID.

For each of the NetPoint-provided plug-ins described in this section, a table is provided which includes information about the plug-in, its parameters, and how it is used.

The following explanations apply to these tables:

- Parameters for all plug-ins are case-sensitive. You must enter them exactly as they are shown in the tables.
- Parameters not labeled as mandatory in the tables are optional.

Credential Mapping Plug-In

Your authentication scheme must provide the functionality implemented by the credential mapping plug-in. It must map the user’s credentials to information in the LDAP directory server. If you do not use the NetPoint-provided credential_mapping plug-in, you must create a custom plug-in that performs the same task. Table 8 gives the parameters you use for the credential mapping plug-in.

There are two parameters important to credential mapping that you must support if you provide your own plug-ins. Both are required:

- **obMappingBase**—The search base against which the search for user credentials begins
- **ObMappingFilter**—The search criteria for the filter

Both parameters are used to map to the user’s credentials to a Distinguished Name (DN) in the directory.

Table 8 Credential Mapping Parameters

Name	credential_mapping	
Purpose	Maps user-provided information to a valid DN in the directory	
Result	If one DN (not zero and not more than one) matches the specified criteria, authentication continues. The obMappingBase and obMappingFilter parameters are added to the list of credentials and the internal uid is set to the DN. The plug-in fails if zero or more than one DN is returned.	
Parameters	obMappingBase	Base DN in the LDAP search.
	obMappingFilter	Filter in the LDAP search: • This parameter is mandatory. • The value specified for this parameter is used to filter for categories of end users.
	obdomain	Used only to authenticate a user against an Active Directory Forest when the challenge method is basic.
	EnableCredentialCache	Turns the credential mapping cache on or off in the credential_mapping plug-in. By default, the credential mapping cache is turned off. Oblix recommends that you accept the default for the credential mapping cache.

Filtering Inactive Users

You can add the obuseraccountcontrol parameter to the obMappingFilter parameter used for the credential mapping plug-in. This makes it possible to filter two categories of users:

- Users who have been added to your directory server, but who have not been activated in NetPoint
- Users who have been deactivated from NetPoint, but who are still in your directory server

Here is an example of an obuseraccountcontrol term to filter out the above two categories of users:

```
(|(!(obuseraccountcontrol=*))
      (obuseraccountcontrol=ACTIVATED))
```

If obuseraccountcontrol is ACTIVATED, or there is no value, then inactive users are filtered out. The obuseraccountcontrol parameter must be used with the obMappingFilter parameter. It cannot be specified without obMappingFilter.

Validate Password Plug-In

The validate_password plug-in validates the user's password against the specified directory server for the authentication scheme. For validate_password, the Access Server uses the same directory server against which it performed the credential_mapping plug-in with a successful outcome.

Here is an example of settings for the validate_password plug-in:

```
validate_password
obCredentialPassword="password", obAnonUser="cn=anonymous,
o=Company, c=US"
```

Table 9 describes the validate password plug-in.

Table 9 Validate Password Plug-In

Name	validate_password
Purpose	Validates the user-provided password against the user's password in the directory.
Result	If the user-entered password matches the password in that user's directory entry, authentication continues. If not, the plug-in fails.

Table 10 describes the parameters for the Validate Password plug-in.

Table 10 Validate Password Plug-In Parameters

obCredentialPassword	Specifies the name of the password field. This parameter is mandatory, and it must be listed first.
obAnonUser	Specifies a DN that is authenticated with any password. This DN must map to a user profile, preferably with restricted access. There may be multiple obAnonUser parameters for a single plug-in. Examples: guest, anonymous.
obCredValidationByAs	When set to true, the Access Server validates passwords using its cache. A user's initial attempt is validated by the directory server. The Access Server caches an MD5 hash of the password and checks the password when subsequent requests are made. If the given and cached password match, the password is considered valid.
obPwdHashTTL	This setting controls the interval during which the Access Server validates passwords by comparing them with a cached password. After the interval, the Access Server returns to the directory server to validate each password. The default value is 1800 seconds (30 minutes).
obReadPasswdMode and obWritePasswdMode	Values can be LDAP or CACHE: • If the value is LDAP, the user's entry is obtained from the directory server for each authentication. • If the value is CACHE, the first authentication is made from the directory server, and afterward from the cache. Values for both parameters (obReadPasswdMode and obWritePasswdMode) should be the same. That is, both parameters should be either LDAP or CACHE. Enables password management for the authentication scheme in the Access Server.

Certificate Decode Plug-In

The certificate decode plug-in extracts the components of the certificate subject's and issuer's Distinguished Name (DN). For each component, the plug-in inserts a credential with a certSubject or certIssuer prefix. For instance, if your certificates have a subject name such as givenName=*somename*, the plug-in adds the

credential certSubject.givenName=*somename* to the credential list.

The following table describes the certificate decode plug-in.

Name	cert_decode
Purpose	Decodes the certificate and extracts the elements of the certificate's subject and issuer DN. This plug-in can be used with the X.509 Cert challenge method.
Result	If the decoding is successful, the elements of the certificate's subject and issuer DN are added to the list of credentials. If not, authentication fails.
Parameters	None

If your certificate is stored in the browser, you can view the certificate details.

The following table lists the OIDs of the attributes that are supported by the Access Server with the corresponding suffix used to retrieve the attribute.

OID	Component lookup name
2.4.5.3	CN
2.5.4.4	SN
2.5.4.5	Serial Number
2.5.4.6	C
2.5.4.7	L
2.5.4.8	ST
2.5.4.9	Street Address
2.5.4.10	O
2.5.4.11	OU
2.5.4.12	Title
2.5.4.13	Description
2.5.4.14	Search Guide
2.5.4.15	Business Category
2.5.4.16	Postal Address

OID	Component lookup name
2.5.4.17	Postal Code
2.5.4.18	Post OfficeBox
2.5.4.19	Physical Delivery Office Name
2.5.4.20	Telephone Number
2.5.4.21	Telex Number
2.5.4.22	Telex Terminal Identifier
2.5.4.23	Facsimile Telephone Number
2.5.4.24	x121 Address
2.5.4.25	International ISDN Number
2.5.4.26	Registered Address
2.5.4.27	Destination Indicator
2.5.4.28	Preferred Delivery Method
2.5.4.29	Presentation Address
2.5.4.30	Supported Application Context
2.5.4.31	Member
2.5.4.32	Owner
2.5.4.33	Role Occupant
2.5.4.34	See Also
2.5.4.35	User Password
2.5.4.36	User Certificate
2.5.4.37	CA Certificate
2.5.4.38	Authority Revocation List
2.5.4.39	Certificate Revocation List
2.5.4.40	Cross Certificate Pair
2.5.4.41	Name
2.5.4.42	Given Name

OID	Component lookup name
2.5.4.43	Initials
2.5.4.44	Generation Qualifier
2.5.4.45	Unique Identifier
2.5.4.46	DN Qualifier
2.5.4.47	Enhanced Search Code
2.5.4.48	Protocol Information
2.5.4.49	Distinguished Name
2.5.4.50	Unique Member
2.5.4.51	House Identifier
2.5.4.52	Supported Algorithms
2.5.4.53	Delta Revocation List
2.5.4.58	Certificate Attribute
2.5.4.65	Pseudonym
1.2.840.113549.1.9.1	E
0.9.2342.19200300.100.1.1	UID

Notice that most of the names are space separated. The following is an excerpt of code used to retrieve these values from an authentication plug-in:

```
sn = pFnBlock->GetCredFn(pInfo->Creds, “cerSubject.Serial
Number”);
```

To view the certificate details

1. Open up an IE browser.
2. Click Tools > Internet Options.
3. Click the Content Tab.
4. Click the Certificates button
5. Double-click your certificate.
6. Click the Details tab.
7. Click the Subject line.

ValiCert Plug-In

The ValiCert Plug-in is used to check the validity of certificates. When you use the ValiCert plug-in in an authentication scheme, do the following:

- In the authentication scheme, specify plug-ins in the following order:
  - cert_decode plug-in
  - authn_valicert plug-in
  - credential_mapping plug-in
- Install the Valicert Validation Authority (VA) Certificate in a Base 64 encoded format in *AccessServer_install_dir*/access/oblix/config/valicert/va_cert.cer file.
- Install the CA Chain Certificate in a Base 64 encoded format in a file located in *AccessServer_install_dir*/access/oblix/config/valicert.
- Create a file named ca_certs.lst to store the CA Chain Certificate. Save this file in *AccessServer_install_dir*/access/oblix/config/valicert.

  AccessServer_install_dir is the directory where the Access Server is installed.

Table 11 describes the ValiCert plug-in.

Table 11 ValiCert Plug-In

Name	authn_valicert
Purpose	Accesses ValiCert's validation authority with the necessary certificates to perform validation of the client certificate.
Result	If ValiCert returns a successful validation, authentication continues. If not, authentication fails.
Parameters	vaURL—The URL to the ValiCert Validation Authority (VA). This parameter is mandatory.

Caching Validated Passwords to Increase Performance

By default, the directory server validates user passwords. To increase performance, you can use the Access Server to validate passwords after the first time they are validated by the directory server.

For this purpose, you must:

- Include the validate_password plug-in in an authentication scheme
- Set the plug-in's obCredValidationByAS parameter to true

When the obCredValidationByAS parameter is set to true, the Access Server caches an MD5 hash of a user's password after it is validated by the directory server.

The next time the user attempts to access a resource within the same policy domain, the user's password is compared with the cached password. If the two match, the given password is validated and the user is granted access to the requested resource.

Another parameter, obPwdHashTTL, controls the length of time the Access Server validates passwords. The default is 1800 seconds (30 minutes). You can change this value. When the specified length of time elapses, the password validation function returns to the directory server.

Here is an example of settings for these parameters that allow the Access Server to validate passwords for 100 seconds:

```
validate_password obCredentialPassword="password",,
obCredValidationByAS="true", obAnonUser="cn=anonymous,
o=Company, c=US", obPwdHashTTL="100"
```

Adding Plug-Ins and Managing Them

The steps of an authentication scheme include one or more plug-ins. Before you can add plug-ins to a step, you must add to the authentication scheme all of the plug-ins to be used for any of its steps.

Before you can add plug-ins to an authentication scheme, you must:

- Define the plug-ins

  For information about defining plug-ins, see "Plug-Ins for Authentication" on page 165.

- Disable the authentication scheme

  For details, see "Enabling and Disabling Authentication Schemes" on page 164.

The first time you add plug-ins to an authentication scheme, the NetPoint Access System creates a default step that includes all of them. If you are using an authentication scheme from a release of NetPoint prior to the NetPoint 6.5 Access System, the Access Server creates a default step for the authentication scheme containing all of its plug-ins.

Viewing Plug-Ins for an Authentication Scheme

You can list an authentication scheme's plug-ins at any time. For example, you may want to list the plug-ins to see ones already added to that scheme before you add others. The plug-ins list displays the names and parameters of the plug-ins already added to the authentication scheme. The list may include any NetPoint-provided and custom plug-ins previously added to the scheme.

To view the list of plug-ins for an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The List All Authentication Schemes page appears.
3. In the List All Authentication Schemes page, click the scheme for which you want to display a list of plug-ins.
4. Select the Plugins tab.

   The plug-ins for an Authentication Scheme page appears, as illustrated below.

Note: It is possible to have more than one credential_mapping plug-in. To help you identify each plug-in, the parameter definitions for each plug-in are displayed in the status bar at the bottom of the browser.

Adding a Plug-In to an Authentication Scheme

When you add a plug-in to an authentication scheme, you specify the name of the plug-in and its parameters. You can add the same plug-in more than once to an authentication scheme if each instance of the plug-in has different parameters. Each instance of a plug-in with unique parameters appears as a separate plug-in in the list.

Use the following task to add a plug-in to an authentication scheme, whether you are adding it to a new scheme or an existing one.

To add plug-ins to an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the link for an authentication scheme.

   The Details for an Authentication Scheme page appears.
4. Click Modify.
5. Disable the authentication scheme if it is enabled.

   a) Select the No radio button for Enabled.

   b) Answer Yes to the confirmation prompt.

   c) Click Save on the Modifying Authentication Scheme page.

   The Details for Authentication Scheme page appears.

   For details, see “Enabling and Disabling Authentication Schemes” on page 164.
6. Select the Plugins tab to display the plug-ins for this authentication scheme.

7. Click Modify

   The Plugins for Authentication Scheme page appears, as illustrated below.

8. Click Add.

   The Plugins for Authentication Scheme page appears, as illustrated below. This page includes a drop-down list and a text box for selecting and defining the plug-in to be added. You either select a NetPoint-provided plug-in or enter the name of the custom plug-in in the Plugin Name box.

   To add a custom plug-in, enter the name of the new plug-in in the Plugin Name box, and enter its parameters in the Plugin Parameters box.

A credential mapping plug-in is required for every authentication scheme. You can select the NetPoint-provided plug-in, or you can select a custom one that implements the same behavior. You either select a plug-in from the right-most Plugin Name text box or enter the name of a custom plug-in in the text box.

For details describing the NetPoint credential_mapping plug-in, including requirements your custom plug-in must meet if you provide a replacement, see “Credential Mapping Plug-In” on page 170.

Each parameter can have multiple values.

To add more plug-ins, click the Add button after you finish adding the previous one. Repeat this step for each plug-in that you want to add.

9. Click Save to save the plug-ins you configured (or click Cancel to exit the page without saving the plug-ins).

Be sure to re-enable the authentication scheme after you have completed any changes you want to make to it. For details, see “Enabling and Disabling Authentication Schemes” on page 164. Authentication schemes must be enabled to be available for use in a rule. If a disabled scheme is used in action domains or policies, the resource is not protected.

Deleting Plug-Ins from an Authentication Scheme

You can remove a plug-in from an authentication scheme, but you must first remove the plug-in from any steps of the scheme that include it. Before you remove the plug-in from the steps, and then from the scheme, you must disable the scheme. For explanation of how to disable an authentication scheme, see “Enabling and Disabling Authentication Schemes” on page 164.

To delete plug-ins from an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.
3. Select the name of the authentication scheme whose plug-in you want to delete.

   The Define an Authentication Scheme page appears.
4. Click Modify.
5. Disable the authentication scheme if it is enabled.
   a) Select the No radio button for Enabled.
   b) Answer Yes to the confirmation prompt.

c) Click Save on the Modifying Authentication Scheme page.

The Details for Authentication Scheme page appears.

For details, see “Enabling and Disabling Authentication Schemes” on page 164.

6. Select the Plugins tab to display the plug-ins for this authentication scheme.
7. Click Modify.

   The Plugins for Authentication Scheme page appears, as illustrated below.

8. Select the plug-in that you want to delete by checking the box before the name of the plug-in. To delete more than one plug-in, select each of them.
9. Click the Delete button below the list to delete the selected plug-ins from the authentication scheme.
10. Click Save.

About Chained Authentication Configuration

When a user requests access to a resource protected by an authentication rule, the rule's authentication scheme determines the way in which authentication is to be performed. For chained authentication schemes, this process includes obtaining user credentials and mapping those credentials to a user profile. A chained authentication scheme can be designed to do this and more. For example, instead of limiting the search for a user profile to one directory, a chained authentication scheme can support attempts to map the credentials to a user profile in one directory, another directory, or yet another directory consecutively, until the information is found. It can also include additional processes indirectly related to the authentication process.

Process overview: A simple chained authentication scheme

Step 1—Plug-ins for Directory A map user credentials to one directory server and verify those credentials. If either plug-in of Directory A fails, the step specifies that Step 2 is to be executed.

Step 2—Plug-ins for Directory B map user credentials to another directory server and verify the credentials.

Step 3—If either plug-in of Directory B fails, this step specifies that the plug-ins for Issue Message and Quit are to be executed.

Creating an Authentication Rule Using Chained Authentication

Here is an overview of the process you use to set up chained authentication for a policy domain. This process assumes that the policy domain exists and that the plug-ins to be used have already been defined.

You use the Access System Console for all of the following steps except the last one—creating an authentication rule. To create an authentication rule for a policy domain, you use the Access Manager.

Task overview: Defining and using a chained authentication scheme

1. Define a chained authentication scheme, as described in "Creating an Authentication Scheme" on page 154.

   Before you can create an authentication scheme containing one or more steps, you must first define the scheme. This process includes specifying the challenge method to be used.

2. Add to the chained authentication scheme all of the plug-ins to be used for its steps, as described in “Adding Plug-Ins and Managing Them” on page 178.

   For example:

   a) Select a plug-in to be added from among the ones NetPoint provides by default, or specify existing custom plug-ins.

   b) Specify the parameters for each plug-in as you add it to the scheme. You can add more than one instance of a plug-in to a scheme, each with its own set of parameters.

   c) Repeat this process for as many plug-ins as you want to add to the scheme.

   Note: When you add plug-ins to an authentication scheme, the Access Manager creates a default step and adds them to it.

3. Add the steps of the authentication scheme, as described in “Configuring and Managing Steps” on page 191.

   Before you create a step, consider its purpose within the authentication scheme and in relation to other steps of the scheme.

   Planning for the steps of a scheme and the scheme’s flows are interdependent processes. You can use steps to isolate plug-ins into groups. You can then connect those groups of plug-ins—that is, connect their steps—in different ways, creating different authentication flows.

   Here is how to add a step to an authentication scheme:

   a) Give the step a meaningful name. Well-chosen names are helpful if you rearrange the steps of a scheme.

   b) Add plug-ins to the step.

   c) Arrange the plug-ins in the order in which you want them executed.

   Take into account how the result of a plug-in affects the result of a step for:

   - NetPoint

     If any NetPoint-provided plug-in fails, the step fails.

   - Custom plug-ins

     For details, see “Return Codes for Plug-Ins” on page 166.

   Add as many steps as are necessary to complete the authentication process for your environment.

4. Create the authentication flows of the chained authentication scheme, as described in “Authentication Flows” on page 197.

   Plan the authentication flows of a scheme. Before you configure a scheme’s authentication flows, take the time to plot the actions you want to occur for each step.

   Here is how to create a scheme’s authentication flows:

   a) Determine which step you want to be executed first. Mark it the *initiating step*.

   b) Configure the links for the step:

      - Determine the next step to be executed if the plug-ins of a step cause the step to fail.
      - Determine the next step to be executed if the plug-ins of a step cause the step to succeed.

5. Verify the flows of the chained authentication scheme, as described in “Authentication Flows” on page 197.

   Test the way you configured the flows—that is, the connections between the steps creating flows—to ensure that there are no cycles.

6. Correct the flows of the chained authentication scheme, if necessary, as described in “Authentication Flows” on page 197.

7. Enable the authentication scheme after you are satisfied with its configuration, as described in “Enabling and Disabling Authentication Schemes” on page 164.

8. Create an authentication rule which includes the chained authentication scheme for the policy domain, as described in “Authentication Rules” on page 205.

9. Specify actions for the authentication rule to be taken if authentication fails or if it succeeds based on the rule. For details, see “Authentication Actions” on page 211.

Authentication Steps

An authentication scheme includes one or more steps whose execution order is determined dynamically. Execution of the steps of an authentication scheme begins with the one chosen as the starting, or initiating, one. From the starting step and for each succeeding step, the step to be executed next is determined by the result of the preceding step.

Each step of an authentication scheme contains one or more plug-ins. Plug-ins within a step are executed in the order in which you position them.

At any time, you can change

- The connections between the steps of an authentication scheme.
- The order of a step's plug-ins.

Within a step, if any NetPoint-provided plug-in fails, the Access Server treats the step as if it failed and stops execution of the step at that point.

For information describing how the Access Server responds to a step containing a custom plug-in based on the execution result of that plug-in, see "Return Codes for Plug-Ins" on page 166.

Figure 6 illustrates a prototype for a sample authentication scheme.

Figure 6 Sample Authentication Scheme Prototype

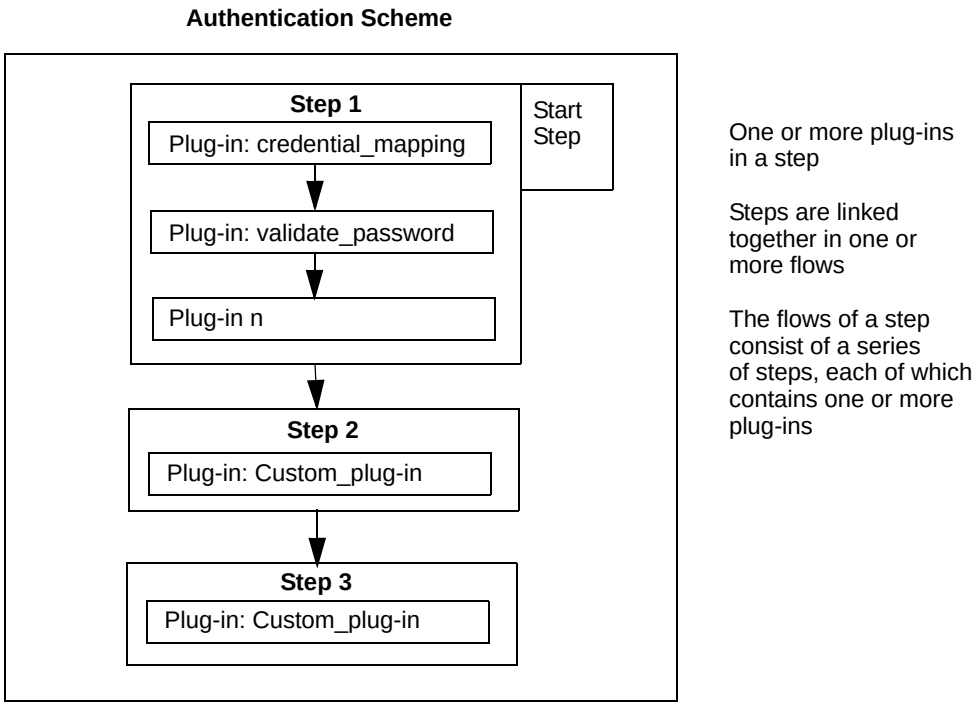


Table 12 summarizes the components of a step.

Table 12 Aspects of a Step

Step Name	A step is a discrete entity. Each step must have a unique name.
Plug-Ins for a Step	A plug-in provides an authentication scheme's functionality. A step can contain one or more plug-ins, but it must contain at least one. The parameters a plug-in can take are specified when the plug-in is added to the scheme, not when it is added to a step.
Number of Steps	An authentication scheme can contain any number of steps, but it must contain at least one.
Connections Between Steps	Steps are connected to form one or more flows of an authentication chain. Because steps are discrete, they can be combined in any order. Connections between steps are established by defining possible authentication flows—or flows of execution—through the authentication chain. See "Authentication Flows" on page 197 for details.
Execution of Steps	Steps are executed in the order in which they occur in a flow of the authentication chain. The plug-ins of steps are executed in the order in which they are positioned in a step's list of plug-ins. The order of plug-ins in a list can be changed. Execution of one step's plug-ins is followed by execution of those of the next step in the authentication flow.

About Single-Step Authentication Schemes

Many authentication schemes are simple enough to require only a single step. In such a case, the step must contain all of the plug-ins required to transact the purpose of the scheme. Because it is the only step in the authentication scheme, the authentication scheme's flow consists of execution of the step's plug-ins.

You can use the Access Manager's authentication feature to create a single step that provides all of the functionality you may require to obtain user credentials, map them to an entry in the directory server, authenticate the user, and so forth.

It is easy to create and manage an authentication scheme with a single step, and it makes good sense to include all plug-ins in a single step in many cases. For example, a single-step authentication scheme is useful if a group of plug-ins are meant to be executed consecutively and, in the event of failure, you do not care which plug-in causes the step to fail.

Why Separate Plug-Ins Into Steps?

You may want to separate plug-ins into steps because the plug-ins form a set meant to be executed together. Also, combining the plug-ins in a step allows you to use that step in a scheme more than once. You may want to configure it as the next step to be executed for one step if that step fails, and as the next step to be executed for another step if that step succeeds.

You may also find it necessary to separate plug-ins into discrete steps even if two plug-ins form a couple logically. For example, you may want to take this approach if you must know which of two plug-ins caused the authentication process to fail.

There are many cases for which you may want to separate closely related plug-ins into discrete steps. Use of the password management feature offers an example of one case. An organization uses password management to control user access. Based on the number of attempts specified in the password policy, it gives a user a certain number of opportunities to enter the correct password. If authentication fails, the administrator must know why. The administrator must be able to distinguish between the following two events:

- Whether authentication fails because there is no entry for the user in the directories checked.
- Whether authentication fails because the user entered the wrong password each time for the three allowed attempts.

For example, an organization uses two different parts of its directory to store user profile information for its human resources department and for its marketing department. The organization wants to be able to search across both branches of the directory for user profile information to authenticate users. The organization wants the search to begin with the search base for the human resources information and if the user profile information is not found there, continue with the search base for the marketing department.

- Search Base A
  - Includes entries for all human resources department members.
  - Examples:
    - cn=Maurice Breton
    - cn=Alice Smith
- Search Base B
  - Includes entries for all marketing department members.
  - Example:
    - cn=Sonal Kalra
    - cn=Robert Jang

The organization defines the following chained authentication steps:

- Step 1: Credential mapping
  - Success: Execute Step 2
  - Failure: Execute Step 3
- Step 2: Validate password
  - Success: Execute Step 4
  - Failure: Stop

  The validate password plug-in gives the user three attempts to enter a valid password, which is based on a setting in the password policy, before it fails Step 2.
- Step 3: Custom plug-in to check Search Base B
  - Success: Execute Step 2.
  - Failure: Stop
- Step 4: Custom plug-in to do some additional processing
  - Success: Stop, return result.
  - Failure: Stop, return result

Sonal Kalra requests access to a resource protected by this authentication scheme. She enters her username. Here is the process that occurs:

A. The Access Server searches Search Base A for an entry for Sonal Kalra.

- There is no entry (Step 1: failure. On failure, go to Step 3.)

B. The Access Server searches Search Base B for an entry for Sonal Kalra

- An entry with cn=Sonal Kalra is found (Step 3: success. On success, go to Step 2.)

C. Sonal Kalra is prompted for her password

- She enters the wrong password the first time. Step 2: validate password prompts her for her password three times before returning a failure.
- At the second prompt, she enters the correct password (Step 2: On success, go to Step 4)

D. Some additional processing is done, which completes successfully (Step 4: On success: Stop, return result.)

If Sonal Kalra entered the wrong password for each of the three attempts, Step 2: validate password, would return a result of failure, and the authentication process would stop. The Delegated Access Administrator would know why the authentication process failed—not because no user entry was found for Sonal Kalra, but because she entered the wrong password three times.

About the Default Step

The first time you add plug-ins to an authentication scheme, the Access Manager defines a default step that contains all of the plug-ins. You can modify the default step if you want to use it, or you can delete it after you add one or more additional steps to the scheme. An authentication scheme must include at least one step.

Configuring and Managing Steps

After you define an authentication scheme and add plug-ins to it, you can configure its steps. You can modify the steps of an authentication scheme at any time, but you must first ensure that the scheme is not used by any active policy domains. You can add plug-ins to a step or remove them from one, or you can delete the step.

Viewing the Steps of an Authentication Scheme

You can view a list of the currently configured steps of an authentication scheme.

To view the steps of an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme whose steps you want to see on the Authentication Management: List All Authentication Schemes page.

   The Details for Authentication Scheme page appears. By default, the General page is displayed.

4. Select the Steps tab.

   The Steps for Authentication Scheme page appears, as illustrated below. This page displays the names of all the steps configured for the scheme. Each step's name is a link, which you can click to display details about the step.

   If you are creating an authentication scheme and have not yet added any steps to it, or if the scheme contains only a single step, this page shows only a step called Default Step, as illustrated above. For details, see "About the Default Step" on page 191

Viewing the Configuration Details for a Step

You can view the details of the current configuration of a step for an authentication scheme any time after it is created.

To view the details for a step

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme containing the step whose configuration details you want to see.

   The Details for Authentication Scheme page appears. By default, the General page is displayed.
4. Select the Steps tab.

   The Steps for Authentication Scheme page appears. This page displays the names of all the steps configured for the scheme.

5. Click the name of the step whose configuration you want to see.

   The Steps for Authentication Scheme page appears again, as illustrated below, this time showing the details for the selected step.

Adding a Step to an Authentication Scheme

To add a step to a scheme, you name the step and add to it the plug-ins that provide the step's functions. For steps with more than one plug-in, the order in which you position the plug-ins in the step determines their execution order. The highest order plug-in—the one at the top of the list—is executed first.

When you add a plug-in to a step, it is placed at the bottom of the list of active plug-ins. You can rearrange the order of plug-ins in a step.

To add a step to an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme on the Authentication Management: List All Authentication Schemes page.

   The Details for Authentication Scheme page appears. By default, the General page is displayed.

4. Select the Steps tab.

   The Steps for Authentication Scheme page appears.

   If you are creating an authentication scheme and have not yet added any steps to it, this page shows only a step called Default Step. For details, see “About the Default Step” on page 191.

5. Click Add.

   The Modify an Authentication Step page appears, as illustrated below. Note that although this page is titled Modify, it is used to add a step as well as to modify the content of an existing one.

6. Enter a unique name for the step in the Step Name text box.
7. From the list of available plug-ins, select the plug-in to be added to the step and click Add.

   The name of the plug-in appears in the Active Plugins scroll box.

   Repeat this step for as many plug-ins as you want to include in the step.
8. To reposition plug-ins within the step, select the plug-in in the list of active plug-ins, and click the appropriate arrow key to move the plug-in up or down in the list.
9. Click Save.

Modifying a Step

You can modify existing authentication steps. For example, you may want to upgrade a step's plug-ins, replacing one with another, or you may want to add new plug-ins to a step to extend or change its function. You may also want to remove plug-ins which are no longer used.

To add plug-ins to an existing step, remove them from it, or change their order

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the left navigation pane.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme whose step you want to change.

   The Details for Authentication Scheme page for that scheme appears.
4. Select the Steps tab.

   The Steps for Authentication Scheme page appears.
5. Click the name of the step that you want to modify.

   The Steps for Authentication Scheme page appears, showing the plug-ins and parameters for the step.
6. Click Modify

   The Modify an Authentication Step page appears, as illustrated below.

7. Change the plug-ins in the step in any of the following ways:

- To add a plug-in to the Active Plugins list, select the plug-in from the Available Plugins list and click Add.
- To remove a plug-in from the active list, select the plug-in from the Active Plugins list, and click Delete.
- To change the order of plug-ins in the Active Plugins list, select the plug-in you want to move. Use the arrow keys to move the plug-in up or down in the list.

8. Click Save to save the step after you are satisfied with the changes.

Deleting a Step

You can delete one or more steps from a scheme. An authentication scheme must have at least one step.

To delete a step from an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.
3. Click the name of the authentication scheme whose step you want to delete.

   The Details for an Authentication Scheme page for that scheme appears.
4. Select the Steps tab.

   The Steps for Authentication Scheme page appears.
5. Select the step that you want to delete.

   Select the check box for each step that you want to delete, if you want to delete more than one.
6. Click Delete.

Authentication Flows

An *authentication flow* is a path of execution through steps of an authentication scheme, or, for single-step authentication schemes, through their plug-ins.

Either of the following kinds of authentication schemes has an authentication flow:

- A single-step authentication scheme

  The authentication flow of an authentication scheme containing a single step consists of the flow of execution through that step's plug-ins in the order in which they appear in the step. For a description of single-step schemes, see "About Single-Step Authentication Schemes" on page 188.

- A chained authentication scheme

  The authentication flows of a chained authentication scheme consist of the execution of the plug-ins of one step after another in a flow. A chained authentication scheme can have one flow or many flows. The execution order of the authentication scheme's steps can vary to create different possible authentication flows, depending on the outcome of each step in a flow.

  For each step of an authentication scheme, you configure the next step to be executed based on the result of the current one. If the current step fails, the step you configured for that step's failure result is executed next. If the current step succeeds, the step you configured for that step's success result is executed next. The plug-ins of any of the steps of an authentication flow are executed in the order in which they appear in the step.

You use the following means to configure the steps of an authentication scheme to produce various possible flows:

- Mark a step as the initiating step of the chained authentication scheme.

  All possible flows of a chained authentication scheme begin with the same step. Each authentication scheme can have only one step designated as the initiating step.

- Specify the next step to be executed if the present step fails or if it succeeds.

  This mechanism allows you to configure different flows of a chain, each of which is determined by the result of the current step.

- Use the Stop terminator.

  Any step of a chain may be followed by the Stop terminator. You can specify that execution is to stop if a step fails or if it succeeds. For either case, you set the failure condition or the success condition of the step to Stop. You can terminate execution after a step absolutely by setting both conditions of the step's result to Stop.

  You may want execution to terminate under more than one condition for the flows of a chained authentication scheme, depending on the possible flows. Stop indicates that a flow has ended and no other steps of the authentication chain are executed.

Authentication Flows Example

An administrator for a company wants to organize the plug-ins used for authentication into steps so that she can more easily control the order in which they are executed. The administrator wants the result of execution of one plug-in to determine the next plug-in to be executed. If the plug-ins were to be executed in order, it would not be necessary to separate them into steps. The company uses the four plug-ins identified in Table 13 for its authentication process.

Table 13 Plug-Ins for Authentication Flow Example

Plug-In	Use
Plug-in 1: credential_mapping	NetPoint-provided credential mapping plug-in
Plug-in 2: validate_password	NetPoint-provided password validation plug-in
Plug-in 3: custom_pluginA	do_what_I_want: A
Plug-in 4: custom_pluginB	do_what_I_want: B

The administrator has determined that she wants to combine her authentication plug-ins into steps that allow her to define the following authentication flows:

- If plug-in 1 is successful (credentials mapped to user entry)

  Execute plug-in 2 (validate the user's password)

- If plug-ins 1 and 2 are successful (user's credentials map and user's password is valid)

  Execute plug-in 4 (do_what_I_want:B)

- If plug-in 1 or 2 fails (either the user's credentials cannot be mapped to an entry or the user's password is invalid)

  Execute plug-in 3 (do_what_I_want:A)

- If plug-in 3 succeeds (do_what_I_want: B),

  Execute plug-in 4 (do_what_I_want:B)

The administrator creates the three steps identified in Table 14.

Table 14 Steps for Authentication Flow Example

Step	Plug-Ins Used	Step Result
Step 1	Plug-in 1 and Plug-in 2	Succeeds if both Plug-in 1 and Plug-in 2 succeed. Fails if either of the plug-ins fails.
Step 2	Plug-in 3	Succeeds if Plug-in 3 succeeds. Fails if Plug-in 3 fails.
Step 3	Plug-in 4	Succeeds if Plug-in 4 succeeds. Fails if Plug-in 4 fails.

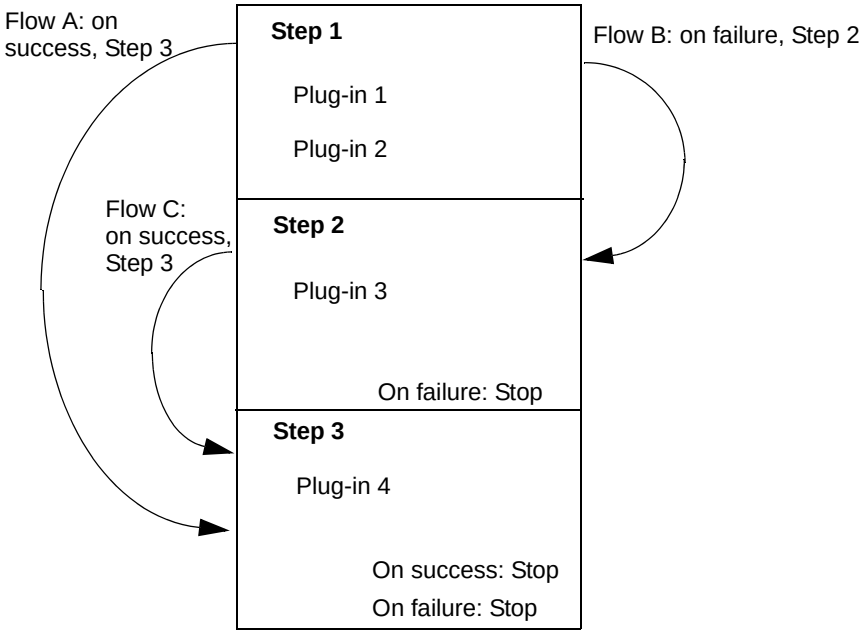
She combines the plug-ins in Table 13 with the steps in Table 14 to create the desired authentication flows. Table 15 shows the steps of the authentication scheme and which step is executed next if the step succeeds or if it fails.

Table 15 Outcome of Steps for Authentication Flow Example

Step	On success, execute. . .	On failure, execute. . .
Step 1	Step 3	Step 2
Step 2	Step 3	Stop
Step 3	Stop	Stop

Figure 7 provides a diagram of the authentication flow table

Figure 7 Illustration of Authentication Flow in Table 15



Viewing the Flows of an Authentication Scheme

At any time after you configure the authentication flows for an authentication scheme, you can look at the configuration by selecting the Authentication Flow tab. The Flow of the Authentication Scheme page shows the current configuration.

After you add a step to an authentication scheme, by default the Access Manager assigns the Stop terminator to the On Success Next Step and On Failure Next Step result conditions of each step. The Flow of the Authentication Scheme page shows this default configuration until you modify it.

To view the configuration of an authentication flow

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.

3. Select the name of the authentication scheme whose authentication flows you want to view.

   The Details for Authentication Scheme page for the authentication scheme appears.

4. Select the Authentication Flow tab.

   The Flow of the Authentication Scheme page appears.

Configuring and Modifying the Flows of an Authentication Chain

After you add steps to an authentication scheme, you can configure the possible flows of execution through the steps. You use the Flow of the Authentication Scheme page to configure the On Success Next Step and On Failure Next Step result conditions for each step.

At any time, you can use the same page to modify the flows of the authentication scheme. You can change the links between steps in a chain to correct cycles or to redirect flows.

Before you modify the flows of an authentication scheme, you must first disable the scheme. For details about disabling the scheme and the effect it has on active policy domains, see “Enabling and Disabling Authentication Schemes” on page 164.

To configure the flows of an authentication scheme

1. From the Access System Console, click the Access System Configuration tab.
2. Click the Authentication Management link in the side navigation bar.

   The Authentication Management: List All Authentication Schemes page appears.

3. Select the name of the authentication scheme whose authentication flows you want to configure.

   The Details for Authentication Scheme page for the authentication scheme appears.

4. Select the Authentication Flow tab.

   The Flow of the Authentication Scheme page appears. For existing steps, this page shows the connections between steps of the chain. If there is only one step for the scheme, it appears here.

5. Click Modify

   The Flow of the Authentication Scheme page with modifiable entries appears, as illustrated below. The page shows the names of the scheme's steps. For each step, the page includes lists from which to choose the next step to be executed if the current one succeeds or if it fails.

6. Choose the step to be used as the initiating step by selecting the radio button for the step in the Initiating Step column.

   Only one step can be configured as the Initiating step.

7. For each step in the Step Name column, complete the following:

   a) In the list under the On Success Next Step column, select the next step to be executed if the present one succeeds.

   b) In the list under the On Failure Next Step, select the next step to be executed if the present one fails.

   If you want execution to terminate after a step is completed, select the Stop terminator. You can use Stop for success of a step or for failure of a step.

   Both selection lists show the names of all steps configured for the chained authentication scheme.

8. After you are satisfied with the configuration, click Verify Flow to determine if it contains cycles.

   See "Verifying and Correcting Cycles in an Authentication Flow" on page 203 for details.

9. Click Save after you have determined that there are no cycles in the flows.

Verifying and Correcting Cycles in an Authentication Flow

Because the flows of an authentication chain can be complex, it is possible for a chain to include cycles.

After you define how the steps of a scheme are connected, you can click the Verify Flows button to check the configuration for cycles before you save it. The Verify Flows button is the rightmost one at the bottom of the Flow of the Authentication Scheme page, as illustrated below.

If the authentication flow's configuration contains cycles, the Access Manager identifies the offending flow on the All Flows in the Chained Authentication Scheme page.

You cannot save the configuration until you correct the cycles. If you attempt to save an authentication flow's configuration without having verified it first, the Access Manager automatically checks the configuration to ensure that none of its flows contain cycles.

Although the Access Manager verifies the authentication flows to check for cycles, it is good practice to plot the flows of a complex authentication scheme well before you configure them.

To correct flows containing cycles after they are reported on the All Flows in the Chained Authentication Scheme page, you use the Flow of the Authentication Scheme page.

The All Flows in the Chained Authentication Scheme page shows all of the configured flows, depicting them in the following way:

- Flows without cycles are shown in black.
- Flows with cycles are shown in red.

To correct an authentication flow containing a cycle

1. Note the reported flow and its offending step in the flows display of the All Flows in the Chained Authentication Scheme page, as illustrated in the example of a flow with cycles below.

   If the verification process reports more than one flow containing cycles, note and correct all of them.

2. Click Back on the All Flows in the Chained Authentication Scheme page, which reports the offending flow.

   The Flow of the Authentication Scheme page appears.

3. Correct the problem within the flow that contains the cycle.

   "Configuring and Modifying the Flows of an Authentication Chain" on page 201 describes the process to use to create authentication flows. Follow this process to modify the connections between the offending steps.

4. Click Verify Flow.

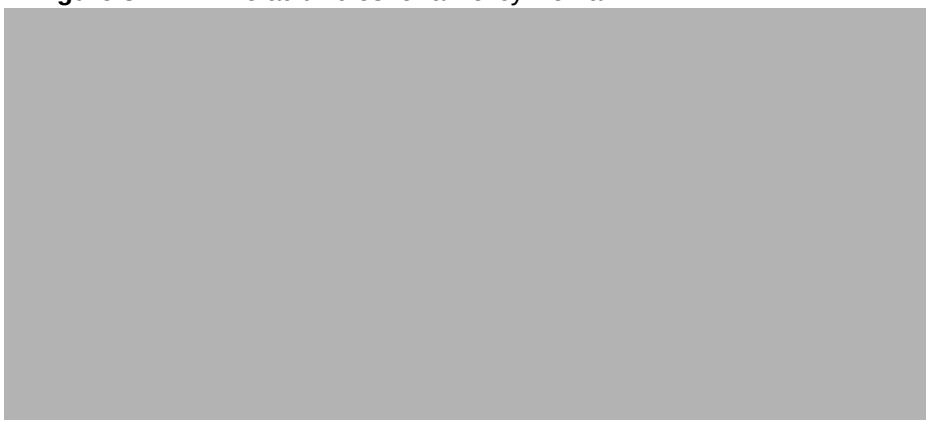
   If the verification results show more flows with cycles, continue to correct the flow.

5. After all problems causing the cycle are resolved, click Save.

Authentication Rules

Each policy domain must include a single default authentication rule, and each policy in a domain can include an authentication rule specific to the policy. If a policy does not include an authentication rule, it inherits protection by the default authentication rule established for the entire policy domain. Figure 8 illustrates conceptually the set of default rules for a policy domain, among which is an authentication rule. In this example, no policies have been created yet for the policy domain.

Figure 8 Default Rules for a Policy Domain



An authentication rule includes an authentication scheme that specifies the kind of authentication required to verify a user's identity, the directory server to be checked for user information, and so on. For details, see "Authentication Schemes" on page 152.

Whenever a user requests access to a resource protected by an authentication rule, the user must authenticate to NetPoint using the challenge method specified by the rule's scheme.

Delegated Access Administrators can create authentication rules for the policy domains and their policies for which they have administrative rights.

Creating an Authentication Rule for a Policy Domain

For each policy domain, you must define a single default authentication rule.

To create a default authentication rule for a policy domain

1. Launch the NetPoint Access System and select Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. For the selected policy domain, select the Default Rules page.

   If there is an authentication rule already configured for the policy domain, the Authentication Rule page appears showing the definition of the rule.

   There can be only one default authentication rule per policy domain. If there is an existing default authentication rule, you must delete it before you can add a new one. For details, see “Deleting a Policy Domain’s Authentication Rule” on page 207.

3. Click the Add button on the Authentication Rule page.

   The General page for the Authentication Rule appears.

4. Enter a Name for the default authentication rule.

5. Enter a Description for the default authentication rule.

6. Select an authentication scheme.

   The drop-down list includes enabled authentication schemes created by the Master Access Administrator. To add new schemes, if required, see “Creating Authentication Schemes” on page 153. Authentication schemes that are disabled do not appear in the list.

7. Click Save.

Modifying an Authentication Rule for a Policy Domain

You can modify the authentication rule for any policy domain for which you have administrative rights, including any policy domain that you have created.

To modify a policy domain’s authentication rule

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. Click Default Rules.

   The General page for the Authentication Rule tab appears showing the current configuration for the rule.

3. Click Modify.

   The General page, whose fields you can modify, appears, as illustrated below.

4. Change the Name, Description, and Authentication Scheme fields as necessary.
5. Click Save to save your changes (or click Cancel to exit the page without saving).

Deleting a Policy Domain's Authentication Rule

Because a policy domain can have only one authentication rule, you must delete the existing rule before you can add a new one.

To delete a policy domain's authentication rule

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. Click Default Rules.

   The General page for the Authentication Rule tab appears showing the currently configured rule.

3. Click Delete.

   Answer Yes to the prompt, to confirm the deletion.

Creating an Authentication Rule for a Policy

For any policy domain, you can create special policies for groups of resources within the domain. All resources of a policy domain are protected by its default authentication rule unless the resource is covered by a policy containing a different authentication rule. You define an authentication rule for a policy just as you would for a policy domain, but you define the rule in association with the policy.

If an authentication rule exists for the policy and you want to replace it, you must delete the rule before you can create a new one. See “Deleting an Authentication Rule for a Policy” on page 210.

To create an authentication rule for a policy

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. For the selected policy domain, select the Policies tab.

   The Policies page appears listing all of the existing policies, if any, as illustrated below.

3. Select the Policy for which you want to add an authentication rule by clicking the link for that policy.

   The General page showing the configuration for the policy appears.

4. Select the Authentication Rule page.

5. Click Add.

   The General page for defining an authentication rule appears.

6. Enter a Name for the default authentication rule.

7. Enter a Description for the default authentication rule.

8. Select an authentication scheme.

   The drop-down list shows s the authentication schemes created by the Master Access Administrator. To add new schemes, if required, see “Creating Authentication Schemes” on page 153.

9. Click Save.

Modifying an Authentication Rule for a Policy

You can modify the authentication rule for a policy within a policy domain for which you are granted administrative rights and for a policy within a policy domain that you have created.

To modify a policy’s authentication rule

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General tab is highlighted and the page shows details for the selected policy domain appears.

2. Select the Policies tab to display a page listing all existing policies.

3. From the list of policy names, select the Policy whose authentication rule you want to modify.

   The Policies General page appears showing the configuration for the policy.

4. Select the Authentication Rule tab.

   The Authentication Rule General page appears, listing the definition of the authentication rule, as illustrated below.

5. Click Modify.

   The Authentication Rule General page form appears enabling you to edit the information using text boxes and a list.

6. Modify the definition of the policy's authentication rule as necessary, changing its name, description, or the authentication scheme it includes.

7. Click Save.

Deleting an Authentication Rule for a Policy

You can delete the authentication rule for a policy within a policy domain if you are granted administrative rights, and for a policy within a policy domain that you have created.

To delete a policy's authentication rule

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. For the selected policy domain, select the Policies tab.

   The Policies page appears listing all of the existing policies.

3. Select the Policy whose authentication rule you want to delete.

   The General page showing the configuration for the policy appears.

4. Select Authentication Rule.

   The General page showing the definition of the authentication rule appears.

5. Click Delete.

   Answer Yes to the confirmation prompt.

Authentication Actions

You can configure an authentication rule that returns actions to be taken depending on the outcome of the rule. You can also specify actions to be taken depending on whether authentication succeeds or fails.

Actions allow you to pass user profile information for the user requesting the resource to other applications or to redirect the user's browser to another site. The use of actions is optional.

Actions are used in the following ways:

- If an allow result is returned, the actions of the rule that determined the allow result are taken.
- If a deny result is returned, the actions of the rule that determined the deny result are taken

---

Note: When configuring actions for an Active Directory forest using ADSI, be sure the administrative account is set to *AD Domain*/administrator in the Windows Directory Security: Authentication and Access Control manager.

About Kinds of Actions

Actions allow you to

- Redirect the user's browser to another URL.

  You can redirect URLs from the Access Server to an AccessGate or a WebGate.

- Pass information about the user to downstream applications in the same NetPoint policy domain or a different one.

  Using HTTP header variables or cookies, you can use actions to pass the following kinds of information:

  - User profile information
  - A user's DN
  - Static text strings

See “About the Use of HTTP Header Variables and Cookies” on page 212 for details about using header variables to pass information to downstream applications.

Note: Redirection and use of header variables are mutually exclusive.

About the Use of HTTP Header Variables and Cookies

Consider the 4K size limit of the HTTP header when you use HTTP header variables and cookies to pass information to downstream applications. This HTTP header size limit includes all cookies, server variables, and environment variables—that is, all of the content of the HTTP header. There is no constraint on the number of individual elements an HTTP header can contain if the content does not exceed the 4K limit. Therefore, when assessing the amount of available space in the HTTP header, take into account the byte size of the data used by NetPoint and other applications. For example, if NetPoint and other applications combined used 1K in the HTTP header, you would have 3K for your data.

Passing Information Using Actions

You can use actions for many purposes. The following table provides some examples of how to use actions.

Table 16 Examples of how to use actions

Task	Example
Personalizing the end user’s interaction with the receiving application.	You can use an authentication action to send the user’s name to a downstream application. The application could use the name to greet the user with a personalized message when the user logs in.
Passing information in a header variable.	You can use a header variable: • To pass membership information • To pass information about a user for purposes of single sign-on For SSO to work, the target application must be able to use the variable.
Redirecting users to a specific URL upon failure or success of the attempt to authenticate.	You can use redirection to send the user to another location. For example, you can redirect a user to your portal page following authentication through your custom form.

Important: Redirection and use of header variables are mutually exclusive.

Actions and Header Variables

You can use HTTP header variables as vehicles for passing static values or identifying attributes. Authentication actions occur once during a user's session—when the user logs in. Header variables passed as authentication actions are not persistent during a user's session.

Note: A header variable is limited to 4KB. This size includes cookies, the server, and environment variables.

How Caching Header Variables Affects their Availability

If a header variable's value is changed, the new value is not available until the Access Server cache is refreshed.

There are two cache timeout parameters that affect header variables:

- **User Cache Timeout**—When an attribute in the header variable is obtained from the directory, it is placed in the user cache. If the value of this attribute changes and there is no user cache flush request for that user, the Access Server does not know about this change until the user cache timeout occurs. At this point, the Access Server retrieves the data again from the directory.
- **Policy Cache Timeout**—For policy data, if a user changes the return attribute in an action, and this change does not reach the Access Server (for instance, if a cache flush failed), the Access Server does not know about this action until the policy cache timeout limit is reached.

Ways Different Web servers Handle Header Variables

Web servers process header variables differently. This variability affects how you must implement header variables in your applications.

Here are some examples:

- Netscape/iPlanet Web servers precede NetPoint variables with the string, HTTP:
  - If you define a variable called HTTP_CN, Netscape/iPlanet produces a variable called HTTP_HTTP_CN.
  - When you write an application that needs to read a header variable, the application must look for a variable called HTTP_HTTP_CN and not HTTP_CN.

- Microsoft IIS expects header variables to be defined with a dash, not an underscore. You would enter HTTP–CN, not HTTP_CN.

  The receiving application must read the variable as if it had an underscore. It looks for HTTP_CN, not HTTP–CN.

- The Lotus Domino Web server cannot pass NetPoint header variables.

For information about how to use header variables for various servers, refer to your Web server's documentation.

Using Actions for Redirection

You can use actions to redirect a user from the target page to a different one. You can use form-based authentication to send users to another page when authentication succeeds, rather than to the originally requested URL. This is a popular use of redirection.

For example, a user might request www.dirac.com/spin/index.htm.You could create a custom form to be used to challenge the user. After the user is authenticated based on information they enter in the custom form, you might redirect the user to your main portal page. You could redirect the user's browser instead of sending the user to the resource requested initially. To do so, you enter the portal page URL in the redirect field when you configure the action.

Note: If you redirect a user upon authentication success or failure, NetPoint does not pass the header variables. Oblix considers passing header variables on redirection a security risk.

You may want to redirect a user upon authentication failure if you want them to see a more informative Web page than the standard HTTP-404-Page Not Found.

Using Form-Based Authentication Instead of a Plug-In

Instead of implementing a plug-in to prompt your users for two levels of authentication information, you may want to use two consecutive form-based authentication screens.

You can design two HTML forms, each of which has text fields for users to enter credentials. You define credential mapping for each login form. You present the user consecutively with the two HTML form-based screens. When the user clicks on the form's submit button, the form data is intercepted and processed by WebGate before it is posted to the Web server. The WebGate searches the directory for profiles with attributes matching the form credentials.

For example, Arete Airlines provides employees with personal flight benefits accrued over time. The IT department of the airline has implemented a form-based authentication system to present two consecutive HTML form-based screens to the user. Each form requests a different kind of information for user authentication, and each form has its own security level:

- **First screen**—Prompts the user for Employment Area and Organization Number.

  This form-based authentication method may have a low security level, such as 1, because many people know the information.

- **Second screen**—Prompts for the user's Personal Information Number (PIN).

  This form-based authentication method may have a high security level, such as 3, because the information is private, identifying the user exclusively.

Process overview: Form-based authentication from the user's perspective

1. The user clicks a link on the company human resources site for employee flight benefits.
2. The application presents the user with the first form-based HTML page, prompting the user for department information.
   - If authentication succeeds for the first screen input, the user is presented with the second form-based HTML page.
   - If the user's PIN is authenticated, the user is granted access to the resource.

For more information about form-based authentication, see "Form-Based Authentication" on page 349.

Custom Actions

If you want to customize the action taken in response to an authentication result, you can create your own actions.

To implement custom actions, you create a plug-in to be called in response to the authentication result.

You can design your external code to execute any number of actions. Some examples are:

- Accessing a relational database using required parameters
- Passing the username of the user who has successfully been authorized for a resource
- Adding optional parameters that define a user's access

Setting Authentication Actions

You use the Actions page of the authentication rule page to create authentication actions. Actions are optional. You can specify them for authentication failure, authentication success, or both.

You can redirect header variables only to Web servers known or protected by NetPoint. Header variables are not redirected outside of NetPoint.

For example, you could enter HTTP_HELLO in the Name field and cn in the Return Attribute field. In this case, NetPoint sends a value to the Web server in the HTTP header called HTTP_HELLO, including the user's common name for the cn attribute. An application could then examine this HTTP header variable. It could then display the value using application code to personalize an interface to include the user's name.

If the attribute contains multiple entries, such as phone numbers, NetPoint returns them as a single string in colon-separated format. End users must parse the individual values themselves.

Enter obmygroups:*ldap_url* to return only specific groups a user is a member of. For example, enter obmygroups:ldap:///o=company,c=us??sub?(group_type=role) to return all of the groups in the DN that the user is a member of and that have the group_type set to role. To return all of the groups a user is a member of, enter obmygroups in the Return Attribute field.

To set authentication actions for a policy domain

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. Click Default Rules > Authentication Rule tab.

   The General page of the Authentication Rule tab appears.

3. Click Actions.

4. Click Add.

   The Actions page for the authentication rule appears, as illustrated below.

5. Specify the actions to be taken in response to successful authentication of the user in the Authentication Success text boxes.

   For details, see the next procedure.

6. Specify in the Authentication Failure text boxes the actions to be taken if authentication of the user fails.

   For details, see the next procedure.

7. Determine when you want Access Server chaches to be updated.

   - Select Update Cache if you want all Access Server caches to be updated *immediately* with information about this new prefix.
   - If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.

8. Click Save to save your input and return to the previous page (or click Cancel to return to the previous page without saving).

To define authentication actions for a policy domain

1. In the Redirect To fields (for both Authentication Success and Authentication Failure), type the complete path to a URL where the end user's browser is sent after the request is received.

   Note: Header variables can be redirected only to Web servers known to or protected by NetPoint. Header variables are not redirected outside of NetPoint.

   Examples:

   - For authentication success, use the following URL to redirect the end user to a portal index page.

     http://mycompany.com/authnsuccess.htm

   - For authentication failure, use the following URL to redirect the end user's request to an error page or a self-registration script.

     http://mycompany.com/authnfail.htm

2. In the Return Type field, specify the method NetPoint uses to send the value to the AccessGate. The method you specify must be recognized by your AccessGate.

   An AccessGate can use these two types of methods:

   - `headervar`
   - `cookie`

   If you are using a client written with the Access Server API, you can pass any alphanumeric string as the type and the client can interpret it.

   For details about HTTP header variables, see "About the Use of HTTP Header Variables and Cookies" on page 212 and "Actions and Header Variables" on page 213.

   Note: If you leave the Type field blank, and then click + to add another field (or click Save), NetPoint uses headervar as the default.

3. In the Name field, enter a variable name that defines your return value or return attributes, such as REMOTE-USER to return the UID.

   Your applications must be configured in advance to accept the variables you enter in these fields.

4. In the Return Value field, enter the value that must be assigned to the associated Name variable when the user is authenticated.

5. In the Return Attribute field, enter the LDAP attributes included in the response from the requesting user's Profile.

   Click the + or – icons to add or remove fields as needed.

   Note: If the returned value contains a special character (such as \ or :), these characters are escaped with a backslash (\ ). The obUniqueid special attribute returns the DN.

6. After you define the actions, return to "To define authentication actions for a policy domain" on page 218 to complete configuration of actions for the policy domain.

Defining Actions for a Policy's Authentication Rule

For every policy, you can define actions for that policy's authentication rule to be taken in response to successful authentication of a user or failure to authenticate a user. Actions are optional. You can specify them for authentication failure and authentication success, or both.

To set authentication actions for a policy

1. Launch the NetPoint Access System and click Access Manager > My Policy Domains > *policy_domain*

   The General page for the selected policy domain appears.

2. For the selected policy domain, select the Policies page.

   The Policies page appears listing all of the existing policies.

3. Select the Policy for which you want to define authentication rule actions.

   The General page showing the configuration for the policy appears.

4. Select the Authentication Rule page.

   The General page showing the definition of the authentication rule appears.

   If you are defining the rule, see "Creating an Authentication Rule Using Chained Authentication" on page 184.

5. Click Actions.

6. Click Add.

7. Specify in the Authentication Success text boxes the actions to be taken in response to successful authentication of the user.

   For details, see "To define actions for a policy" on page 220.

8. Specify in the Authentication Failure text boxes the actions to be taken if authentication of the user fails.

   For details, see the next procedure.

9. Determine when you want Access Server chaches to be updated.
   - Select Update Cache if you want all Access Server caches to be updated *immediately* with information about this new prefix.
   - If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.
10. Click Save.

To define actions for a policy

1. In the Redirect To field, type the complete path to a URL where the end user's browser is to be sent after the request is received.

   Examples:

   - For authentication success, use this field to redirect the end user to a portal index page.

     http://mycompany.com/authnsuccess.htm

   - For authentication failure, use this field to redirect the end user's request to an error page or a self-registration script.

     http://mycompany.com/authnfail.htm

2. In the Return Type field, specify the method NetPoint uses to send the value to the AccessGate. The method you specify must be recognized by your AccessGate.

   An AccessGate can use these two types of methods:

   - headervar
   - cookie

   If you are using a client written with the Access Server API, you can pass any alphanumeric string as the type and the client can interpret it.

Auditing Authentication Events

An audit rule causes event-based data to be written to the audit log file. As a Master Access Administrator, you must create a Master Audit Rule in the NetPoint System Console. As a Delegated Access Administrator, you can derive audit rules from the Master Audit Rule for your policy domains and policies, but you cannot create an alternative Master Audit Rule.

There is one audit log per Access Server. You can configure the size of the audit log file and the rotation interval per server. Depending on events, the audit log may contain some duplicate audit entries.

Note: You may direct audit details to a database, as described in *Volume 1*.

Information Logged on Success or Failure

Different information is written to the audit log depending on the outcome of events. A log entry for authentication of a user differs depending on whether the user's identity was established.

Authentication failure can occur if there is no entry in the directory for a user or if a user's credentials are invalid. For example, if there is an entry for the user in the directory, but the user entered an incorrect password (authentication failure), the value for the cn attribute is logged based on the DN in the directory. However, because the entry for the user cannot be confirmed as the correct one, attributes such as givenname are not retrieved from the directory.

About Creating a Master Audit Rule and Derived Rules

You can define audit rules for a policy domain and its policies. Any audit rules you define must be derived from a Master Audit Rule. A Master Audit Rule must be created by a Master Access Administrator. Delegated Access Administrators can derive access rules from the Master Audit Rule, but they cannot create them.

Because you create audit rules for the policy domain and its policies, this chapter does not describe them. For details explaining how to create and define audit rules, see the following sections in the policy domain chapter:

- "Auditing User Activity for a Policy Domain" on page 140
- "About Creating a Master Audit Rule and Derived Rules" on page 221
- "Creating an Audit Rule for a Policy Domain" on page 140
- "Creating an Authentication Rule for a Policy" on page 208

Plug-Ins to Authenticate Users on External Security Systems

NetPoint offers plug-ins that allow you to authenticate users whose information is stored on a Security Bridge server, on Windows NT/2000, or on a SecurID Server. These plug-ins are installed automatically when you install NetPoint.

This section describes the plug-ins for Security Bridge and Windows NT/2000.For information about the SecureID plug-in, see the *NetPoint 7.0 Integration Guide*.

Security Bridge Plug-In

Security Bridge, a product of Security Integration, Inc., is an LDAP interface for OS/390-based repositories. Using the NetPoint authentication plug-in for Security Bridge, authn_securitybridge, you can authenticate users whose information exists in OS/390-based repositories.

Configuration Prerequisites

Before you can configure NetPoint to use Security Bridge, you must install and configure the following:

- Security Bridge LDAP Server
- One of the following OS/390-based repositories:
  - RACF
  - CA-ACF2
  - CA-TopSecret
- NetPoint Access System, including at least one Access Server and one WebGate

The authn_securitybridge.dll (for Windows) or authn_securitybridge.so (for Unix) was installed under the *AccessServer_install_dir*/access/oblix/lib directory.

For more information about installing the Security Bridge server and repositories, refer to the Security Bridge documentation.

Creating an Authentication Scheme for Security Bridge

To authenticate Security Bridge users, you must create an authentication scheme that specifies the Security Bridge plug-in. See Table 17 on page 224 for the complete set of Security Bridge plug-in parameters.

To create an authentication scheme with a Security Bridge plug-in

1. Launch the NetPoint Access System and select Access System Console > Access System Configuration > Authentication Management
2. Click Add.

   The Define an Authentication Scheme page appears.
3. In the Name field, type a name for this scheme, for example, Security Bridge Authentication
4. In the Description field, type a brief description of the scheme, for example, “This authentication scheme requires a user to enter a Security Bridge login and password.”
5. In the Level field, type an integer representing the level of security that this authentication scheme provides, for example, Level: 3
6. In the Challenge Method field, select Form.
7. In the Challenge Parameter field you must add two parameters, form and creds.
   - **form**—The path to the securitybridge sb-login.html file relative to the Web server’s document root.

     For example:

     ```
     form:/securitybridgeforms/sb-login.html
     ```
   - **creds**— A space-separated list of credentials to be passed from the forms to the Access Server.

     For example:

     ```
     creds:login password newpassword
     ```
8. In the SSL Required field, leave No selected.
9. Leave Challenge Redirect blank if there is only one WebGate/Access Server pair. Otherwise, you must redirect to a WebGate that communicates with the Security Bridge authentication Access Server.

10. To add the plug-ins and create the steps and flows for the authentication scheme, see “Adding a Plug-In to an Authentication Scheme” on page 180 and “Adding a Step to an Authentication Scheme” on page 193.

These are the plug-ins you must define for the scheme:

Plugin Name: authn_securitybridge
Plugin Parameters:
username=”uid=%login%, ou=people, o=test.com”,
passcode="password",
newpasscode="newpassword",
ldaphostname="os39029.datadist.com",
ldapport="390",
machine="*machineName*",
formdir="*formDirName*",
securitymode="open",
certfile="c:\Program Files\Netscape\Users*machineName*\cert8.db"

where:

machineName is the machine on which WebGate is located.

formDirName is the directory on WebGate where the Security Bridge forms are located. The default directory is named securitybridgeforms.

Plugin Name: credential_mapping

Plugin Parameter: obMappingBase="o=Company, c=US",
obMappingFilter="(?(objectclass=inetOrgPerson) (uid=%login%))"

Table 17 summarizes available Security Bridge plug-in parameters.

Table 17 Security Bridge Plug-In Parameters

Name	Default	Mandatory/ Optional	Comments
username	uid=%login%, ou=people, o=test.com	Mandatory	
	attribute=%login%, host=myhost, o=mycompany, c=usa	Mandatory	For attributes other than uid—for example cn—use this parameter specification.

Table 17 Security Bridge Plug-In Parameters

Name	Default	Mandatory/ Optional	Comments
password	password	Mandatory	
newpassword	<none>	Optional	Used during password change.
ldaphostname	os39029.datadist.com	Mandatory	IP address of the LDAP server run by Security Bridge to be used for authentication.
ldapport	390	Mandatory	Port number for the server in the previous field.
machine	<none>	Mandatory	Web server machine name.
formdir	<none>	Mandatory	Path relative to the Web server document root.
securitymode	open or SSL	Mandatory	
certfile	<none>	Optional	Location of the cert8.db file that holds all the certificates needed for SSL connections for the LDAP server run by Security Bridge.

Authentication Rule for Security Bridge

Now that you have created an authentication scheme that specifies a Security Bridge plug-in, you can implement the scheme in NetPoint deployments.

To create an authentication rule for Security Bridge

1. Follow the procedures in this chapter for creating an authentication rule.
2. Select Modify, and select the Security Bridge authentication scheme from the Challenge Method list.

After Security Bridge and a NetPoint authentication plug-in have been installed and configured, the authentication process is as follows.

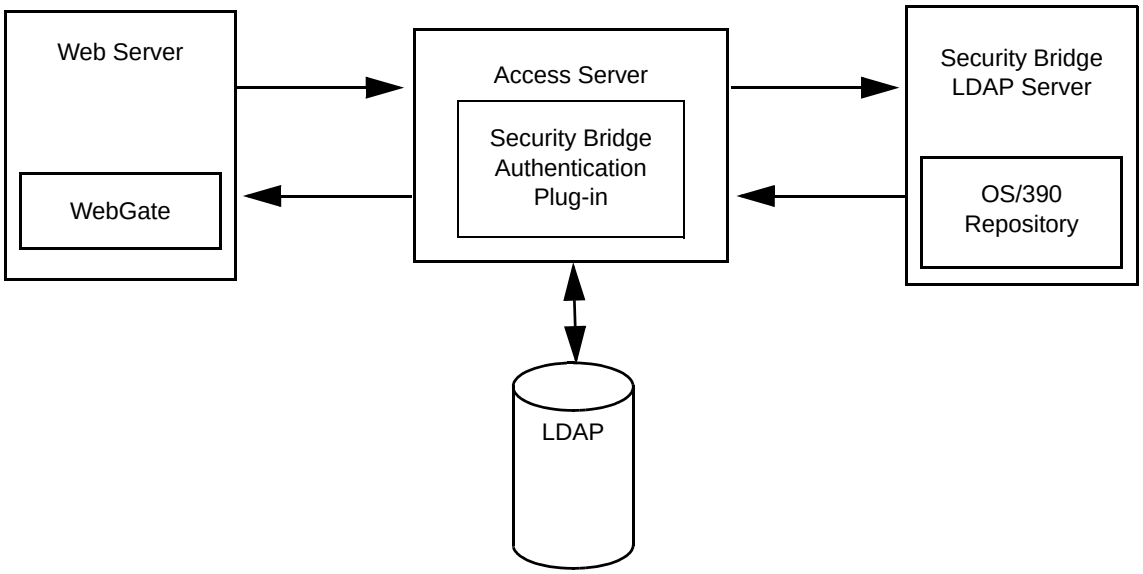
Process overview: Authentication for Security Bridge and NetPoint

1. WebGate intercepts a request to access a resource and determines if the resource is protected.
   - If the resource is protected, WebGate then determines if the user is authenticated.
   - If the user is not authenticated, WebGate issues a form-based challenge and the user supplies the requested login credentials.
2. WebGate then forwards the authentication request to an Access Server.
3. The Access Server processes the request through the Security Bridge authentication plug-in, and the plug-in sends an LDAP bind to the Security Bridge:LDAP Server.
4. The Security Bridge:LDAP Server evaluates the user's credentials stored in the OS/390 repository.

   If the credentials are valid, the request is approved. If not, the request is denied.

Figure 9, "Authentication with a Security Bridge Plug-In" illustrates this process.

Figure 9 Authentication with a Security Bridge Plug-In



Windows NT/2000 Plug-In

Table 18 describes the Windows NT and Windows 2000 plug-in used to authenticate against a Windows domain.

Table 18 Windows NT/2000 Plug-in

Name	authn_windows
Purpose	Authenticates username and password against a Windows NT or Windows 2000 domain.
Result	• If authn_windows returns success, authentication continues. • If not, authentication fails.
Parameters	• **ntusername**—Name of the field containing the username. This parameter is mandatory. • **ntpwd**—Name of the field containing the password. This parameter is mandatory. • **ntdomain**—Name of the field containing the domain.

Securing the ObSSOCookie in an Authentication Scheme

NetPoint implements single sign-on through an encrypted cookie called the ObSSOCookie. You can specify a challenge parameter that ensures the ObSSOCookie is only sent over an SSL connection and prevents the cookie from being sent back to a non-secure Web server.

To secure the ObSSOCookie

1. Create an authentication scheme. See “Creating an Authentication Scheme” on page 154.
2. In the Challenge Parameter field, add another field and specify the following:

   ```
   ssoCookie:secure
   ```

   Note: The Challenge Parameter is case-sensitive. Be sure to enter an uppercase C in ssoCookie.
3. In the SSL Required field, click Yes to ensure the end user is authenticated through an SSL-enabled server.

5 Configuring User Authorization

The NetPoint Access System enables you to protect your resources with policy domains and policies that specify who is authorized to use the resources and who is not allowed to use them, and under what conditions.

This chapter explains authorization and how to configure authorization rules and authorization expressions to meet the requirements for your policy domains and their policies. A policy domain must include an authorization expression among the set of default rules that specify how its resources are protected. Authorization rules are combined to create authorization expressions.

This chapter discusses the following topics:

About NetPoint Authorization

Authorization is the process of determining whether a user has the right to access a requested resource. To protect resources, you define authorization rules which contain conditions. Authorization rules are contained within authorization expressions. A policy domain and a policy can each contain only one authorization expression.

Background Reading

In addition to authorization rules, policy domains and policies also include authentication rules and audit rules, which are described in other chapters of this guide. After you have created your policy domains, you can define their rules and expressions. You can create the authentication rules, authorization rules and expressions, and audit rules for a policy domain in any order. Before you read this chapter, read the following chapters:

- "Configuring AccessGates and Access Servers" on page 33 describes the configuration of AccessGates and Access Servers, which you must do before the policy domains you create can take effect.
- "Protecting Resources with Policy Domains" on page 95 describes how to create and test policy domains and policies, how to define resource types, and how to define audit rules.
- "Configuring User Authentication" on page 149 describes how to create and use authentication schemes and rules.

Introduction to Authorization Rules and Expressions

An authorization rule can contain:

- A condition that specifies who is authorized to access a protected resource. This condition is referred to as the Allow Access condition of the rule.
- A condition that specifies explicitly who is denied access to the protected resource. This condition is referred to as the Deny Access condition of the rule.
- Both Allow Access and Deny Access conditions.

If Allow Access or Deny Access conditions or both are specified and they do not apply to a user, the user is not qualified by the rule. If a user is unqualified by a rule, by default the user is denied access to the requested resource.

To specify who is authorized to use the resource or explicitly denied access, the rule can:

- Identify the users by their user name, role, or an LDAP filter whose criteria the user must satisfy.

- Stipulate the computers from which users can access the resources.
- Set the period of time during which the rule applies.

Additionally, you can set actions to be taken if the rule is evaluated to allow qualifying users access to the resource. You can also set actions to be taken if the rule is evaluated to deny qualifying users access to the resource.

Resources of a policy domain are protected by an authorization expression containing one or more authorization rules.

Authorization expressions include

- Authorization rules that you select from among those that have been defined for the policy domain.
- Operators that you use to combine rules in various ways to provide the kind of authorization protection required for the policy domain.

An authorization expression may consist of a single rule or a group of rules combined to express more complex conditions. For example, you can create an expression which requires that a user meet the Allow Access conditions of two rules to be granted access to the resource. You use the NetPoint Access Manager interface to combine rules in expressions.

This chapter describes the NetPoint Access Manager authorization component, and it explains how it works. It also provides the procedures you use to protect your resources with authorization expressions.

Here is an overview of the steps you follow to create authorization expressions for your policy domains and their policies:

Task overview: Creating authorization expressions

1. Create your policy domain, as discussed in “Protecting Resources with Policy Domains” on page 95.
2. Determine who is authorized to use the resources of the policy domain, and under what conditions, using the “Guidelines for Classifying Users” on page 232. See also “Authorization Rules” on page 233.

   You can give specific users access to the resources. You can also explicitly deny specific users access to the resources. It is not necessary for you to create rules that apply to all of your users—whether to allow them access or to expressly deny them access.

   Some users may not qualify for the conditions of a rule. They may qualify for other rules of the expression, or they may not qualify for the conditions of any rules. If a user does not qualify for the conditions of any of the rules of an expression, by default the user is denied access to the resource.

3. Create all of the authorization rules you need to protect the resources of the policy domain and any of its policies. See “Configuring Authorization Rules” on page 238 for details.

   You create all of these rules at the level of the policy domain. When you create a rule, you include an authorization scheme in it. If you do not plan to use the Oblix Authorization Scheme provided by NetPoint, you must configure one or more custom ones. In this case, you must provide custom plug-ins. For details, see “Authorization Schemes for Custom Plug-Ins” on page 287.

4. Create the authorization expression for the policy domain, which can have only one authorization expression. See “Authorization Expressions” on page 247 for details.

5. Create an authorization expression for each of the policy domain’s policies. See “Authorization Expressions” on page 247 for details.

Note: You must configure an authorization expression to determine if users are permitted to access resources. If no authorization expression is defined, access is denied to the target resources.

Guidelines for Classifying Users

Observe the following guidelines when classifying users:

- Divide the users and groups of users into sets for whom different conditions apply—conditions such as when they can access the resources, the computers from which they must make their requests, and so on. For details, see “About the Contents of an Authorization Rule” on page 235.

  If some users fall into more than one category—for example, a user in the marketing group belongs to the Teleon project group, a user in the human resources group also belongs to the Teleon group—put the user in both categories. You can require that the user meet the conditions of two rules.

  Note: You do not need to be concerned about users who are denied access to the resources of the policy domain under any conditions. They are denied access by default if none of the rules of an expression qualify them.

- For each category for which you want to create a separate rule, consider the kinds of actions you want to occur if the user is authorized to use the resource or if the user is not authorized to use it as a result of the rule. For example, for one case or the other, you may want the system to return user profile information and pass that information to a downstream application:

  - If the user is authorized to use the resource, you may want to pass the user’s cn (common name) to another application so that the application can present a customized greeting to the user.

- If the user is not authorized to use the resource, you may also want to return information about the user to be used for security purposes. (For information about actions, see “Authorization Actions” on page 276.)

Do this analysis for users and groups: users for whom you want to grant authorization to use the policy domain’s resources; users and groups for whom you want to explicitly deny authorization to use the resources.

If you want to create policies for subsets of resources within a policy domain and protect them with different authorization rules, consider the same information for the policies: who can access the resources of the policy and under what condition; for whom, and under what conditions, you want explicitly to deny access to the resources.

Authorization Rules

An authorization rule specifies information that identifies who can access a resource it protects. It also specifies who is explicitly denied access to the resource. One or more authorization rules are included in an authorization expression for a policy domain or policy.

When a user requests access to a resource protected by an authorization rule included in an authorization expression, information about the user is checked against the rule. If the rule stipulates other kinds of information, such as period of time or time of day the rule applies, that, too, is checked. This process is referred to as *evaluation of the rule.*

The result of evaluation of an authorization rule—in conjunction with other authorization rules, if more than one is included in the authorization expression—determines whether a user is granted access to the requested resource.

At the policy domain level, you create all of the authorization rules to be used for a policy domain or any of its policies. You combine these rules to create authorization expressions. For details about authorization expressions, see “Authorization Expressions” on page 247.

This section describes authorization rules, and how to create and manage them. It includes the following topics:

About Allow Access and Deny Access Conditions

An authorization rule specifies the following two types of primary conditions:

- A condition referred to as Allow Access that grants the user access to the resource.
- A condition referred to as Deny Access that denies the user access to it.

When a user is said to qualify for an authorization rule, it does not mean that the user is authorized to use the resource protected by the rule. A user is said to qualify for a rule if the user meets a condition of the rule:

- If the user meets the Allow Access condition, the user qualifies for the Allow Access part of the rule.
- If the user meets the Deny Access condition, the user qualifies for the Deny Access part of the rule.
- If the user satisfies neither the Allow Access nor the Deny Access conditions, the rule is said to be unqualified for that user. You can also think of this as the user not qualifying for the rule. If evaluation of a rule results in an unqualified user, the user is denied access to the resource based on that rule.

For authorization expressions that contain more than one rule, a user may qualify for none of the expression’s rules, one of the rules, or for the conditions of more than one rule. In any case, it is the result of evaluation of the expression—all of its rules and how they are combined—not any one rule, that determines whether a user is allowed or denied access to a resource.

Reuse of Authorization Rules

A policy domain can have only one authorization expression, which can include all of the authorization rules necessary to express the protection requirements for its resources. Each of the policies a policy domain contains can have its own authorization expression.

Any of the authorization rules you define for a policy domain can be used for the policy domain and for any of the policies it contains in the following ways:

- It can appear in more than one authorization expression.
- It can appear in a single authorization expression more than once.

For information about authorization expressions, see “Authorization Expressions” on page 247.

About the Contents of an Authorization Rule

An authorization rule contains the following information:

- **General Information**—An authorization rule has a name and a description, and it can be enabled or disabled. See “Configuring Authorization Rules” on page 238 for details.
- **Allow Access**—The Allow Access condition of an authorization rule specifies the end users and groups of users who are allowed access to a resource protected by the rule. See “Setting Allow Access” on page 240 for details.
- **Deny Access**—The Deny Access condition of an authorization rule specifies the end users and groups of users who are explicitly denied access to a resource protected by the rule. See “Setting Deny Access” on page 242 for details.
- **Timing Conditions**—An authorization rule can be configured to include a value that restricts access to a resource within a period of time, such as 9:00 a.m. to 5:00 p.m. on week days for one group of users and 10:00 a.m. to 4:00 p.m. for another group of users. See “Setting Timing Conditions” on page 243 for details.
- **Actions**—For either result of an authorization rule—whether its evaluation results in authorization success or authorization failure for a user requesting access to a protected resource—an associated set of actions can be specified to be taken in response to the result. For example, the Access System can return a header variable to be passed to a downstream application. The following list describes the kinds of actions you can specify:
  - Redirection of the user’s browser to another URL.
  - Static values and user profile identity values passed in HTTP header variables or cookies.

  See “Authorization Actions” on page 276 for information about actions.

About Authorization Rule Evaluation

When information about a user requesting access to a protected resource is checked against the conditions of an authorization rule, and the user qualifies for one of the conditions of the rule, that rule is evaluated to produce one of the following results:

- Authorization Success

  In this case, the user succeeds in gaining access to the requested resource. This result is associated with the Allow Access condition of the rule.

- Authorization Failure

  In this case, the user fails to gain access to the requested resource. This result is associated with the Deny Access condition of the rule.

Evaluation of a rule can produce neither result if the user requesting access to the protected resource is not mentioned in the Allow Access or the Deny Access conditions of the rule. In this case, the evaluation of the rule is said to be inconclusive, and the user is denied access to the rule.

Working with Authorization Rules

This discussion provides details about configuring and managing authorization rules:

Displaying a List of Configured Authorization Rules

You may find it useful to display a list of authorization rules before you define a new one.

To display a current list of authorization rules

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.

   The General page for the policy domain appears.

2. Select the policy domain whose authorization rules you want to display.

3. Select the Authorization Rules tab for the policy domain.

   The Authorization Rules page appears, as illustrated below, showing the list of authorization rules configured for the policy domain.

Configuring Authorization Rules

To configure an authorization rule, you define its general information, you set its Allow Access and Deny Access conditions, and you define actions for the rule, if any. This section describes how to configure general information for a rule.

You can specify general information about an authorization rule to identify the rule, to specify its authorization scheme, to enable or disable the rule, and so forth. Some of the information you can configure is optional.

You must specify an authorization scheme for every authorization rule you define. You can use the Oblix Authorization Scheme provided by NetPoint or you can select a custom authorization scheme, if any are configured. For details, see "Authorization Schemes for Custom Plug-Ins" on page 287.

You create all of the authorization rules to be used for a policy domain or any of its policies at the policy domain level.

To define an authorization rule

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.
3. Select the Authorization Rules tab.

   A page appears listing existing authorization rules for the policy domain.

   Note: If you are creating a policy domain, you do not see any configured authorization rules.

4. Click Add.

   The General page for the authorization rule appears, as illustrated below.

5. Specify a name for the authorization rule and, optionally, a brief description of it in the following text boxes:

   - **Name**—A name for this authorization rule
   - **Description**—A brief description of this authorization rule

     For example, for an authorization rule that includes a custom authorization scheme, you could explain the function the custom plug-in provides.

6. Select Yes from the Enabled list to enable the authorization rule or No to disable it.

   Select Yes if you want the authorization rule to be activated as soon as you click Save. Enabling an authorization rule makes it available for inclusion in an authorization expression. The rule is disabled by default.

   After an authorization rule is used in an authorization expression, you cannot disable it until it is removed from all of the expressions that use it.

7. For Allow takes precedence, select one of the following:

   - **Yes**—If you want the Allow Access condition to take precedence over the Deny Access condition.
   - **No**—If you want the Deny Access condition to take precedence over the Allow Access condition.

   If you configure Allow Access and Deny Access conditions for a rule, use this option to specify which condition of the rule should be honored if the user qualifies for both of a rule's conditions.

8. Determine when you want Access Server chaches to be updated.

   - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with information about this new prefix.
   - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.

9. Click Save.

   The General page appears displaying the information you specified.

10. Select the authorization scheme to include in the authorization rule.

    If the Master Access Administrator has not created custom authorization schemes, the only scheme available is the Oblix Authorization Scheme.

11. Click Add.

    The General page for an authorization rule appears.

Setting Allow Access

The Allow Access part of an authorization rule defines users and groups who are authorized to use the protected resource.

To set Allow Access

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.

   If you are creating a policy domain, you do not see any configured authorization rules.

4. Select the authorization rule whose Allow Access conditions you want to set.
5. Click the Allow Access tab.
6. Click Add (or Modify if they exist).
7. Specify the users and groups who are allowed to access resources protected by this rule using the People, Role, Rule, and IP Address controls as indicated below.

   Note: These options are alternatives. An end user or group specified in *any* of these fields is allowed access.

   a) **People**—Click Select User to select by user name

- Use the Search facility to display configured users.
- Click Add before the name of each user who is allowed to access resources protected by this rule.

b) **Role**—Select No Role to prevent users from being selected based on roles or select Anyone to allow anyone access to the protected resources.

c) **Rule**—Enter an LDAP filter that specifies the users and groups who are allowed to access the protected resources using the plus and minus buttons to add new filters and delete existing ones.

d) **IP Address**—Enter the IP addresses of computers whose users are allowed access.

Except where noted, NetPoint supports the following conventions for IP addresses in Access System and Access Manager:

- An explicit address, such as 192.2.2.2

- An address with a wildcard, but the wildcard must be the last entry, such as 192.2.2.*, 192.2.*, or 192.*

NetPoint does not support:

- An address in which a wildcard is not the final entry. For example, 192.128.*.2 is not supported.

- An entry of all wildcards, such as ***.*.*.*

If you entered an IP address using a format that is not supported, the error message “Invalid IP address entered” appears.

For the IP Address fields, click the plus and minus buttons to add new IP addresses and delete existing ones.

8. Determine when you want Access Server chaches to be updated.

- **Immediately**—Select Update Cache to update all Access Server caches *immediately* with information about this new prefix.
- **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.

9. Click Save.

Setting Deny Access

The Deny Access part of an authorization rule specifies the users and groups who are denied the right to use the resources protected by this rule.

To set Deny Access

1. Launch the NetPoint Access System and select the Access Manager.
2. From the Access Manager, select My Policy Domains, then click on the policy domain that you want to see.

   If you are in the process of defining the rule and have configured the rule's general information, you do not need to retrace this path.
3. Select the Authorization Rules tab.

   A page appears listing authorization rules for the policy domain.

   If you are creating a policy domain, you do not see any authorization rules.
4. Select the authorization rule for which you want to set Deny Access conditions.
5. Click Deny Access, then click Add (or Modify if they exist).
6. Specify the users and groups who are *denied* to access resources protected by this rule using the People, Role, Rule, and IP Address controls as indicated below.

   Note: These options are alternatives. An end user or group specified in *any* of these fields is denied access.

   a) **People**—Click Select User to select by user name
      - Use the Search facility to display configured users.
      - Click Add before the name of each user who is denied to access resources protected by this rule.

   b) **Role**—Select No Role to prevent users from being selected based on roles or select Anyone to deny anyone access to the protected resources.

   c) **Rule**—Enter an LDAP filter that specifies the users and groups who are denied to access the protected resources using the plus and minus buttons to add new filters and delete existing ones.

d) **IP Address**—Enter the IP addresses of computers whose users are denied access.

Except where noted, NetPoint supports the following conventions for IP addresses in Access System and Access Manager:

- An explicit address, such as 192.2.2.2

- An address with a wildcard, but the wildcard must be the last entry, such as 192.2.2.*, 192.2.*, or 192.*

NetPoint does not support:

- An address in which a wildcard is not the final entry. For example, 192.128.*.2 is not supported.

- An entry of all wildcards, such as ***.*.*.*

If you entered an IP address using a format that is not supported, the error message “Invalid IP address entered” appears .

For the IP Address fields, click the plus and minus buttons to add new IP addresses and delete existing ones.

7. Determine when you want Access Server chaches to be updated.
   - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with information about this new prefix.
   - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.
8. Click Save.

Setting Timing Conditions

Use the Timing Conditions option to set the time periods when the authorization rule is in effect. For example, you may want the rule to remain in effect only during business hours, Monday through Friday. If you do not set a timing condition, by default the authorization rule is always in effect. Take into account that both of the rule’s conditions—its Allow Access and its Deny Access conditions—remain in effect for the specified time period.

To set a timing condition

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.

3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.

   If you are creating a policy domain, you do not see any authorization rules.

4. Select the authorization rule for which you want to set timing conditions.

5. Click Timing Conditions.

   The next screen either lists existing Timing Conditions or reports that there are no Timing Conditions configured for the authorization rule.

6. Click Add (or Modify if they exist).

   The Timing Conditions page appears.

7. Select either Greenwich Mean Time or Local time on Web server:

   - **Greenwich Mean Time**—A standard for universal time. If you use Greenwich Mean Time, this authorization rule is in force at the same time throughout the world.

     Use this option if you want this rule to be in force at the same time for your globally-dispersed workforce.

   - **Local time on Web server**—Indicates that users outside the server's time zone could be denied access.

     For example, if the server is located in New York, and the timing conditions do not allow access after 5 P.M., West Coast users would be denied access starting at 2:01 P.M.

     Note: If you want to restrict hours for users in various time zones, do not use this option. Instead, create a separate authorization rule that gives West Coast users access until 8 P.M. Eastern Time, and so forth.

8. Select a Start Date and End Date.

   Note: If you select the — option, for the Start Date, then this rule effectively does not have a Start Date. If you select the — option, for the End Date, then this Rule effectively does not have an End Date.

9. Select a Start Time and End Time:

   - You cannot choose *only* a Start Time or End Time. If you specify a Start Time, you *must* choose an End Time.

     By default, the Start Time and End Time fields are set to —, which means this rule does not have a Start Time and End Time. It is then in effect 24 hours per day.

   - When choosing a Start Time and End Time, you must make a selection for all three fields (hours, minutes, seconds). If you do not, the Start Time and End Time are invalid.

10. Select the Months of the Year, Days of the Month, and Days of the Week for which this rule is valid.:

    Note: To select a single item (for example, a month) click to select it. To select more than one, hold down the Shift key as you select additional items in the same list. If you select the — option, this rule is in effect everyday.

11. Select Update Cache if you want all AccessGate and Access Servers caches to be updated *immediately* with information about these timing conditions.
12. Click Save.

Viewing General Information About a Rule

You may want to view general information about an authorization rule before you decide to modify the rule or use the rule in an authorization expression.

To view the general information for an authorization rule

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.
4. Select the authorization rule whose general information configuration you want to see.

   The Name, Description, Enabled status, and Allow takes precedence as the status defined for the rule appears.

Modifying an Authorization Rule

You can modify the authorization rules for a policy domain at any time. However, it is good practice to disable a rule before you modify it.

To modify an authorization rule

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.
4. Select the authorization rule that you want to modify.

5. Click Modify.

   The General page with editable text boxes appears.

6. Verify that the Enabled status box is blank to ensure the rule is disabled before modifying information.

7. Modify the general information as required, and any of the following:

   - **Timing Conditions**—Click the tab, and follow the instructions for defining them.
   - **Actions**—Click the Actions tab and follow the instructions for defining actions in “Authorization Expressions” on page 247.
   - **Allow Access** or **Deny Access**—Click the appropriate link, and follow the instructions for defining the rules.

8. Click Save.

Deleting an Authorization Rule

You cannot delete an authorization rule that is used in an authorization expression for the policy domain or any of its policies.

To delete an authorization rule

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that you want to see.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.

4. Select the check box for each rule that you want to delete.
5. Click Delete.

Authorization Expressions

In some cases, a single authorization rule is all that is required to protect the resources of a policy domain or a policy. You can configure the rule to identify who is allowed access to the resources it protects, who is denied access to them, and under what conditions these controls apply—when they apply and from which computer, for example. An authorization rule does not need to cover all users in its Allow Access and Deny Access conditions. Users who do not qualify for any of the conditions of the rule and who request access to a resource protected by the rule are, by default, denied access to the resource.

For other cases, it may be necessary to configure many authorization rules to protect resources with complex restrictions imposed on different users. For example, you may want to define a policy that includes many authorization rules, a part of any one of which a user must meet to qualify for access to a protected resource (or to qualify for denial of access to it). You may also want the same policy domain to specify more than one condition a user must meet. For example, you may require that the user meet two conditions—such as belonging to one group and using a computer assigned a specific IP address—to be granted access to the resource. To define the complete authorization conditions required for the resources you want to protect, you form an *authorization expression*. The NetPoint Access Manager provides an interface that makes it easy for you to form authorization expressions. You must create a default authorization expression for the policy domain, but you can also create an authorization expression for a policy within the domain.

This section describes authorization expressions, and how to create and manage them. It includes the following topics:

About the Contents of an Authorization Expression

Within an authorization expression, you can define all of your authorization requirements for a set of resources, whether those resources are for a policy domain or one of its policies.

An authorization expression includes:

- One or more authorization rules

- The operators used to combine the rules

You can define only one authorization expression for a policy domain—the default authorization expression—and one authorization expression for each of a domain's policies. To create an authorization expression, you use any of the authorization rules defined for the entire policy domain. Figure 10 illustrates aspects of an authorization expression and its place within configuration of a policy domain.

Figure 10 Authorization Expression

An authorization expression is always evaluated from left to right. The rules of an expression can be grouped using operators, and how they are grouped has a bearing on the outcome of the overall evaluation of the expression.

You can use two operators to combine the rules of an expression: AND and OR. You combine authorization rules to create authorization expressions that can include the following types of conditions:

- **A Compound Condition**—Specifies more than one condition for which a user must qualify, either to be granted access to the requested resource or explicitly denied access to it, depending on the rest of the expression. You use the AND operator for this purpose. See "Authorization Rules Used in Example Scenarios" on page 251.
- **A Complex Condition**—Specifies two or more alternative conditions any of which a user must meet, either to be allowed access to the requested resource or denied access to it, depending on the condition and its relationship to the rest of the rules of the expression. You use the OR operator for this purpose. See "Authorization Rules Used in Example Scenarios" on page 251.

See “About Evaluation of the Rules of an Expression” on page 250 for details explaining how grouping of the rules of an expression using AND and OR is interpreted.

About Authorization Expression Evaluation

Evaluation of an authorization expression can result in the following three conditions:

- **Authorization Success**—In this case, the user succeeds in gaining access to the requested resource. This result is associated with the Allow Access condition of the expression.
- **Authorization Failure**—In this case, the user fails to gain access to the requested resource. This result is associated with the Deny Access condition of the expression.
- **Authorization Inconclusive**—In this case, the rules of the expression produce conflicting results, and the user is denied access to the resource.

Status Codes for an Inconclusive Result

An expression can return a result of Inconclusive, in which case the NetPoint Access System returns a major status code of Deny and a minor status code of Inconclusive.

The major status code of Deny is returned for Inconclusive results to maintain compatibility with previous releases of the system. The minor status code of Inconclusive is available to NetPoint systems to allow those systems to distinguish between true Deny results and Deny results returned because of an Inconclusive state.

An authorization expression result of Deny differs from an authorization expression result of Inconclusive even though the user is denied access to the resource in both cases. An application written to run with NetPoint can interpret the two status codes for an Inconclusive result and use the additional information for other purposes. For example, the application can then invoke other authorization engines instead of denying the user access to the resource.

About Evaluation of the Rules of an Expression

An authorization expression can contain a mix of compound conditions and complex conditions which determine whether a user can access a resource protected by the expression. When a user requests access to a protected resource, the user's information is checked against the rules of the expression.

The interplay between user information assessed against the rules of an expression, the position of the rules in the expression, and the way in which the rules are combined in the expression allows for a wide degree of variety. An authorization expression is exercised to different extents depending on these variables—that is, some of its rules might not be evaluated.

Precedence and Position—The Access Server processes the rules of an expression in the following way:

- **Precedence of Operators**—The AND operator takes precedence over the OR operator in regard to how rules of an expression are combined.

  That is, if an expression contains three or more rules combined in some way with the AND operator and the OR operator, the Access Server always associates the rules on either side of the AND operator with it first, and then it combines the rules using the OR operator.

  For example, given the following authorization expression,

  ```
  R1 OR R2 AND R3
  ```

  internally the Access Server creates the following grouping by default:

  ```
  R1 OR (R2 AND R3)
  ```

  The Access Server goes through the entire expression making these groupings based on AND taking precedence over OR before it evaluates the user's information against the rules.

  For details about operators, see "Authorization Rules Used in Example Scenarios" on page 251.

  Note: You can override the default way in which operators are interpreted by using parenthesis to enforce new groupings. For details, see "About the Use of Parenthesis" on page 261.

- **Position of Rules in an Expression**—The Access Server evaluates an expression from left to right.

  You do not assign to an authorization rule its priority among other rules. It would not be possible to reuse authorization rules if you assigned to each of them an evaluation priority. Rather, you position rules in an expression from left to right—which is the order in which they are evaluated—and you use operators to combine them. For details about operators, see "Authorization Rules Used in Example Scenarios" on page 251.

- **Use of Parenthesis to Override Default Precedence**—You can use parenthesis to override the default way in which the Access Server groups the rules of an expression. The Access Server continues to evaluate the rules of an expression from left to right, but it assesses the rules within the couplings and groups you create through use of parenthesis. See “About the Use of Parenthesis” on page 261.

About the Definitive Result of an Authorization Expression—The Access Server evaluates the rules of an expression until it can produce a definitive result. Evaluation of an authorization expression may produce a definitive Allow Access result, a Deny Access result, or an Inconclusive result.

For example, a user qualifies for the Allow Access condition of Rule 1, the Deny Access condition of Rule 2, and the Deny Access condition of Rule 3 of the following expression.

(Rule 1 AND Rule 2) OR Rule 3

In this case, evaluation of Rule 3 produces a definitive result of the expression, and the user is denied access to the resource. Neither Rule 1 nor Rule 2 has any bearing on the outcome of the expression because they produce conflicting results as part of an AND condition. Because Rule 3 is part of an OR condition, it stands on its own. If the user satisfies the rule’s Allow Access or Deny Access condition, then Rule 3 defines the outcome of the expression.

For Rule 2 to be responsible for the definitive result, the user must qualify for either both the Allow Access conditions or both the Deny Access conditions of Rule 1 and Rule 2. In this case, Rule 3 would not be evaluated because evaluation of Rule 1 and Rule 2 would produce a definitive result. Therefore, evaluation of Rule 3 would be unnecessary.

Authorization Rules Used in Example Scenarios

Table 19 contains examples of authorization rules that, if defined at the policy domain level, could be used in authorization expressions for the domain and any of its policies. The example authorization rules in Table 19 show only one condition of a rule—either its Allow Access condition or its Deny Access condition—not the full authorization rule.

An authorization rule need not specify both an Allow Access condition and a Deny Access condition, or either one alone. It can specify either condition, both conditions, or none. Table 19 identifies example authorization rules which are used in example scenarios throughout the rest of this chapter.

Table 19 Example Authorization Rules and Their Conditions

Authorization Rule	Condition
Rule 1	Allow anyone from the marketing department group access to the requested resource.
Rule 2	Allow anyone using a computer with the IP address 192.168.2.123 access to the requested resource.
Rule 3	Allow anyone from the human resources department group access to the requested resource.
Rule 4	Allow anyone from the Teleon project group access to the requested resource.
Rule 5	Deny anyone from the consultants group access to the requested resource.
Rule 6	Deny anyone from the Saber project group access to the requested resource.
Rule 7	Deny anyone using a computer with the IP address 192.168.5.123 access to the requested resource.
Rule 8	Allow anyone from the managers group access to the protected resource.
Rule 9	Allow anyone from the administrative assistants group access to the protected resource.

About the AND Operator

You use the AND operator to form a compound condition which combines authorization rules. Any number of rules can be combined using the AND operator to implement the full scope of conditions a user must meet to satisfy the authorization requirement. However, a user must satisfy the same kind of condition—either Allow Access or Deny Access—of all of the rules of the AND compound condition for the AND clause to produce a definitive result.

An authorization expression can contain more than one coupling or grouping of rules combined using AND. For example, it may contain several AND clauses, one connected to another by an OR operator.

Note: A user may qualify for both the Allow Access condition and the Deny Access condition of the same rule. In this case, whichever condition is configured to take precedence is the one that is honored. You configure this setting in the Allow takes precedence field.

Examples of Compound Conditions

The following scenarios use the example authorization rules in Table 19 on page 252 to illustrate compound conditions.

For some of these examples, the Access Manager Authorization Expressions page you use to create the expression is shown. Here is where to find information explaining how to use these pages to create authorization expressions:

- For the steps to follow to create an authorization expression, see “Creating Authorization Expressions” on page 265.
- For information explaining how to use the Authorization Expression interface portion of the Access Manager to create expressions, see “Modifying an Authorization Expression as You Create It” on page 270. These instructions apply both to creating an expression and modifying an existing one.

A Compound Condition Whose Two Authorization Rules Specify Allow Access Conditions—To be allowed access to a resource protected by the following authorization expression, a user must belong to the marketing department group and the IP address of the user’s computer must be 192.168.2.123.

Rule 1 AND Rule 2

A Compound Condition Whose Three Authorization Rules Specify Allow Access Conditions—To be allowed access to a resource protected by the following authorization expression, a user must belong to the marketing department group, the IP address of the user’s computer must be 192.168.2.123, and the user must be a member of the Teleon project group.

Rule 1 AND Rule 2 AND Rule 4

Here is what the expression would look like if you configured it using the Authorization Expression's Expression page.

A Compound Condition Whose Two Authorization Rules Specify Deny Access Conditions—To be explicitly denied access to a resource protected by the following authorization expression, a user must belong to the Consultants group and belong to the Saber project group.

Rule 5 AND Rule 6

About the OR Operator

An authorization expression can include a complex condition containing two or more alternative authorization rules. Authorization rules forming a complex condition are combined using the OR operator. Each of the authorization rules specified by a complex OR condition stands on its own. Unlike compound conditions using the AND operator, the user need qualify for the condition of only one of the authorization rules connected by OR operators.

An authorization expression can contain as many authorization rules connected using the OR operator as are required to express the authorization policy for the resources it protects. You can use the OR operator to connect authorization rules all of which have Deny Access conditions, all of which have Allow Access conditions, or which specify a mix of Deny Access and Allow Access conditions. You can connect single rules to single rules using OR, and you can connect a single rule to a clause containing rules combined using AND.

Examples of Complex Conditions

The following scenarios use the example authorization rules in Table 19 on page 252.

A Complex Condition Whose Two Rules Specify Allow Access Conditions—To be allowed access to a requested resource protected by the following rule, a user must either be a member of the marketing department group or the human resources department group.

Rule 1 OR Rule 3

Complex Condition Whose Three Authorization Rules Specify Deny Access Conditions—To be explicitly denied access to a requested resource, a user must belong to the Consultants group, or belong to the project Saber group, or use a computer with the IP address 192.168.5.123.

Rule 5 OR Rule 6 OR Rule 7

Here is what the expression would look like if you configured it using the Authorization Expression's Expression page.

A Complex Condition with Rules that Specify a Mix of Allow Access and Deny Access Conditions—To be allowed access to a requested resource protected by the following expression, a user must either be a member of the marketing department group or the human resources department group. To be explicitly denied access to a requested resource, a user must belong to the Consultants group or be a member of the Saber project group.

Rule 1 OR Rule 3 OR Rule 5 OR Rule 6

Here is what the expression would look like if you configured it using the Authorization Expression's Expression page.

Compound Complex Expression Scenarios

The following scenarios use the example authorization rules in Table 19 on page 252 to illustrate authorization expressions that contain both compound and complex expressions.

A Complex Condition Authorization Expression with Three Rules—A Delegated Access Administrator forms the following expression:

Rule 1 OR Rule 2 AND Rule 9

Here is what the expression would look like if you configured it using the Authorization Expression's Expression page.

John requests access to a resource protected by this authorization expression. The Access Server evaluates the expression to determine if John meets either of the following conditions that would allow him access to the resource:

- The IP address of John's computer is 192.168.2.123 and he belongs to the Administrative Assistants group (Rule 2 AND Rule 9)
- John is a member of the marking department group (Rule 1)

If parenthesis were used to make explicit the grouping of rules according to how the Access Server evaluates the authorization expression, the expression would look like this:

Rule 1 OR (Rule 2 AND Rule 9)

An expression is evaluated from left to right until a definitive result is produced. John meets the condition of Rule 1, which is followed by the OR operator, so he is granted access to the resource.

A Complex Condition Expression with Four Rules—A Delegated Access Administrator forms the following expression:

(Rule 1 AND Rule 2) AND (Rule 7 OR Rule 8)

Here is what the expression would look like if you created it using the Authorization Expression's Expression page.

Maurice is allowed access to a resource protected by this authorization expression because he satisfies the following conditions:

- He is a member of the marketing department and the IP address of his computer is 192.168.2.123. (Rule 1 AND Rule 2)
- He is also a manager and belongs to the Managers group. (Rule 8)

  The IP address of Maurice's computer is not 192.168.5.123 (Rule 7). However, he is not denied access for this reason because the authorization expression dictates that he meet either Rule 7 or Rule 8, but not both.

A Complex Condition Expression with Six Rules—A Delegated Access Administrator forms the following expression:

Rule 1 OR Rule 2 OR Rule 3 AND Rule 4 OR Rule 5 AND Rule 6

Here is what the expression would look like if you used the Authorization Expression's Expression page to configure it. Notice that the Authorization Expression List box does not show the last rule. To see the last rule, you would have to scroll down. However, the Text Format box wraps the text to show the complete expression.

If parenthesis were used to make explicit the grouping of rules, the expression would look like this:

Rule 1 OR Rule 2 OR (Rule 3 AND Rule 4) OR (Rule 5 AND Rule 6)

Following the order of precedence of AND over OR in regard to how rules are grouped and left-to-right processing of the rules, a user must qualify for one of the following conditions to gain access to the requested resource:

- The first single rule of the complex condition (Rule 1)

  A user who belongs to the marketing department group is allowed access to the resource.

- The second single rule of the complex condition (Rule 2)

  A user whose computer has the IP address 192.168.2.123 is allowed access to the resource.

- The first compound condition (Rule 3 AND Rule 4)

  A user who belongs to the human resources department group and who belongs to the Teleon project group is allowed access.

- The second compound condition (Rule 5 AND Rule 6)

  A user who belongs to the Consultants group and the Saber project group is denied access to the resource.

In its evaluation, the Access Server progresses through the expression until it evaluates a rule that produces the definitive result of the expression. If the Access Server completes evaluation of the expression and the user does not qualify for any of its conditions, the result of the evaluation is Inconclusive. In such a case, because no rules apply to the user, no actions associated with rules are taken. However, the actions configured for the Inconclusive result of the expression are taken. For information about actions, see “Authorization Actions” on page 276. For information about status codes returned for inconclusive results, see “Status Codes for an Inconclusive Result” on page 249.

About the Use of Parenthesis

By default, two rules on either side of an AND operator compose the compound AND condition. Rules on either side of an OR operator are alternatives. When no parenthesis are used to enforce grouping of rules, the AND operator takes precedence over the OR operator.

For example, if no parenthesis were used in the following expression to override the default way in which the rules of the following expression would be evaluated:

```
R1 OR R2 AND R3 OR R4 AND R5
```

the expression would be interpreted in the following way:

```
R1 OR (R2 AND R3) OR (R4 AND R5)
```

You can use parenthesis to override the normal grouping of the rules of an expression, for example, to give precedence to the OR condition over the AND condition.

The following example uses the same expression. In this instance of the expression, parenthesis are used to override the default grouping:

```
(R1 OR R2) AND (R3 OR R4) AND R5
```

Working with Authorization Expressions

Discussions below provide procedures for working with authorization expressions:

Viewing Authorization Expressions

A policy domain can have only one authorization expression. Each of its policies can also have an authorization expression. If an expression has already been defined for either, you can look at its definition at any time.

If an authorization expression exists for the policy domain or for a policy, the Expression page displays the entire authorization expression. If the authorization expression is long, the text is wrapped onto the next line, and so on, to display all of the expression.

An authorization expression includes the content of the expression—its rules and operators—and the configuration for the expression itself.

To view an authorization expression for a policy domain

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain whose authorization expression you you want to see.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression page appears, as illustrated below. This page shows the name of the expression and its value configured for the policy domain.

To look at values configured for the expression:

- Click Duplicate Actions.

  If a duplicate actions policy has been configured for the authorization expression, this section defines how duplicate actions are handled for the policy domain protected by the authorization expression. A policy domain can include one or more policies.

  See “About Duplicate Actions” on page 283 for details.

- Click Actions.

  This section defines the actions configured for this authorization expression.

5. Click Modify to look at the content of the expression.

The configuration for an expression appears on the page used to create the expression or modify it.

To see the actions configured for each rule of an expression, you must check the rule’s configuration. See “Authorization Rules” on page 233.

Viewing the Authorization Expression for a Policy

Each policy has its own authorization expression. You can view it from within the definition of the policy.

An authorization expression includes the content of the expression—its rules and operators—and the configuration for the expression itself.

To view an authorization expression for a policy

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain containing the policy whose authorization expression you want to see.
3. Select Policies.
4. Select the policy whose authorization expression you want to see.
5. Click Authorization Expression.

   The Authorization Expression page appears, as illustrated below. This page shows the name of the expression, *NetPoint Identity Default Authorization rule*.

6. Click Modify to look at the content of the expression.

   The configuration for the expression appears on the page used to create it or modify it.

   Note: To see the actions configured for each rule of an expression, you must check the rule's configuration. See "Authorization Rules" on page 233.

7. **Optional**—Look at values configured for the expression:

   - Click Duplicate Actions to display the section that defines how duplicate actions are handled for the resources protected by this policy. The setting for the policies authorization expression Duplicate Actions overrides that of the policy domain. See "About Duplicate Actions" on page 283 for details.

     The authorization expression for a policy may contain its own duplicate actions setting. In this case, the policy domain's duplicate actions setting overrides the one set for the policy domain.

   - Click Actions to display the section that defines the actions configured for this authorization expression.

Creating Authorization Expressions

The authorization expression for a policy domain applies to all resources of the domain unless those resources are protected by a policy containing an expression.

To create an authorization expression for a policy domain

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain for which you want to create an authorization expression.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression page appears. If there is no defined authorization expression, a message appears, "There is no Authorization Expression defined," as illustrated below.

   Note: If an authorization expression exists, you can only modify its content. To replace it, you must modify all parts of it.

5. Click Add.

   The Authorization Expression page you use to create the expression appears, as illustrated below.

You use the Authorization Expression page, illustrated above, to create an authorization expression.

6. Using steps below, select the authorization rules for the authorization expression and the operators you want to use to combine those rules.

Note: If you want to include the first rule in a parenthetical phrase, click the open parenthesis button before you add the first rule to the expression.

---

a) From the Select Authorization Rule list, select the first rule to be added to the expression, and click Add.

b) If the authorization expression includes more than one rule, select the operator to be used to combine the first two rules.

- For the AND operator, click the And button beside Select Separator.
- For the OR operator, click the Or button beside Select Separator.
- To begin a parenthetical phrase, click the open parenthesis button.
- To close a parenthetical phrase, click the close parenthesis button.

7. Continue to add rules to the authorization expression, and combine them with other rules until you have completed forming the expression to fit your authorization requirements.
8. Select the Duplicate Actions tab in the Authorization Expression page.

   The Duplicate Action Headers page appears, as illustrated below.

9. Click Modify to select the duplicate actions policy. The Duplicate Actions page appears, as illustrated below.

10. Click Select the checkbox and the radio button for the type of Duplicate Actions handling you want.

    The duplicate actions policy you set at the authorization expression level overrides that set at the Access System Console level.

11. Determine when you want Access Server chaches to be updated.

    - **Immediately**—Select Update Cache to update all Access Server caches *immediately* with information about this new prefix.
    - **Later**—If you do not select Update Cache, the Access Server caches are updated when they time out and read new information from the directory server.

    You cannot save an authorization expression that contains syntax errors. When you click Save, the Access Server checks the authorization expression to ensure that it is well-formed. If an authorization expression contains a syntax error—for example, an error occurs if you include an AND or OR operator at the end of the expression—you must correct the error and then save the expression.

12. Click Save.

    After you save the authorization expression, the Authorization Expression view page appears showing the full expression. For details explaining how to

use the features of the Authorization Expression page to create an expression, see “Modifying an Authorization Expression as You Create It” on page 270.

Creating an Authorization Expression for a Policy

The steps you use to create an authorization expression for a policy are the same as those for a policy. For details, see “Creating Authorization Expressions” beginning on page 265. Start with the step that follows step 5, “Click Add.”

To create an authorization expression for a policy

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain containing the policy for which you want to create an authorization expression.
3. Select the Policies page.
4. Select the name of the policy for which you want to create an authorization expression.
5. Select the Authorization Expression tab.

   The Authorization Expression page appears displaying the message “No authorization expression is defined for this policy.”
6. Click Add.

   The Authorization Expression with an active list box, text entry box, and scrollable lists appears.

Modifying an Authorization Expression as You Create It

As you create an authorization expression, you may want to change the way you have combined the rules of the expression. You may change the form of an expression as you create it, for example, to express a different authorization policy or to correct errors.

If the authorization expression contains many components—rules and operators—a scroll bar is displayed at the right side of the authorization expression list box so that you can scroll to bring items into view.

You can modify an authorization expression in either of the following two ways:

- You can modify an authorization expression within the Authorization Expression list box
- You can modify an authorization expression within the Authorization Expression in Text Format box.

Changes you make to an authorization expression in one box are reflected in the other box in the following ways:

- As you form the authorization expression by adding rules and operators to the Authorization Expression list box, the Authorization Expression in Text Format box is automatically updated to reflect the additions and modifications.
- After you make changes to an expression in the Authorization Expression in Text Format box, you must click the Update button for those changes to be reflected in the Authorization Expression list box.

The way some operators are expressed in the Authorization Expression list box differs from how they are expressed in the Authorization Expression in Text Format box. The following table shows the differences.

Table 20 Operators for List Box and Text Format Box

Operator in List Box	Operator in Text Format Box
AND	&
OR	|
(	(
)	)

You use buttons to enter operators in the Authorization Expression List box. You use keys to enter operators in the Authorization Expression in Text Format text box.

Using the Authorization Expression List Box

The Authorization Expression list box displays the authorization rules and the operators that you use to combine them as you select and add rules and operators to form the expression.

Note: As you create an authorization expression using the Authorization Expression list box, the expression content is reflected in the Authorization Expression in Text Format editable text box.

To manipulate the content of an expression in the Authorization Expression list box, you use the following buttons:

- **Modify**—Replaces one rule of an authorization expression with another rule selected from the Select Authorization Rule list.

  To replace one operator with another, you swap the two operators directly by selecting one operator and clicking the button for the replacement operator.

- **Delete**—Deletes any selected item—a rule, an operator, or an open or close parenthesis—from the Select Authorization Rule list.
- **Delete All**—Clears the entire content of the authorization expression.

To replace one authorization rule with another

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain whose authorization expression you want to modify.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
5. Click Modify.

   The Authorization Expression with an active drop-down list, text entry box, and scrollable list box appears.
6. Select the rule to be replaced in the Authorization Expression list.
7. Select the replacement rule in the Select Authorization Rule list.
8. Click the Modify button.

   The old rule in the Authorization Expression list is replaced by the new rule.

To replace one operator with another

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain whose authorization expression you want to modify.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
5. Click Modify.

   The Authorization Expression page with an active drop-down list, text entry box, and scrollable list box appears.
6. Select the operator to be replaced in the Authorization Expression list.
7. Click the button for the replacement operator.
   - To replace the OR operator with the AND operator, select OR in the expression, and click the And button.

- To replace the AND operator in the expression, select it and click the Or button.

The old operator is replaced by the new one in the Authorization Expression list.

To delete an item

1. Navigate to the Authorization Expression list.
2. Select the item to be deleted in the Authorization Expression list.
3. Click the Delete button.

To delete the entire content of an expression

1. Navigate to the Authorization Expression list.
2. Click the Delete All button.

Using the Authorization Expression in Text Format Box

As you form the authorization expression by adding rules and operators to the Authorization Expression list, the Authorization Expression in Text box is updated to reflect the additions and modifications.

You can modify the textual content of an authorization expression directly using the Authorization Expression in Text box.

Entering New Text—To modify the text, you use keyboard or keypad keys and symbols to enter new text or to overtype existing text. (In addition to typing the text, the main difference is that you enter symbols to represent operators.) See Table 20 on page 271 for the symbols to use for operators.

Deleting Text—To delete text from the authorization expression, you use any of the standard approaches you take to handle text in a flat text file.

Updating the Authorization Expression List—To update the list with the changes you made in the Authorization Expression in Text Format text box, click the Update button directly beneath the text box.

Modifying an Existing Authorization Expression

If an authorization expression exists for the policy domain or for a policy, the Authorization Expression view page displays the entire expression. If the authorization expression is long, the text is wrapped onto the next line, and so on, to display all of the expression.

You can modify an authorization expression after it has been used to protect the policy domain or the policy for which it was created.

When modifying an authorization expression, you follow the same procedures you use to create an expression. This section describes how to navigate to the Authorization Expression page you use to modify an expression for a policy domain and for a policy.

To display the page for modifying the authorization expression for a policy domain

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain for which you want to create an authorization expression.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
5. Click Modify.

   The Authorization Expression page with an active drop-down list and two text entry boxes appears.

For the remainder of this process, see the steps of the following procedures for creating and modifying an authorization expression.

- “Creating an Authorization Expression for a Policy” on page 270.
- “Modifying an Authorization Expression as You Create It” on page 270.

To display the Authorization Expression page for a policy to modify the expression

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain containing the policy for which you want to create an authorization expression.
3. Select the Policies tab.

4. Select the name of the policy whose authorization expression you want to modify.
5. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
6. Click Modify.

   The Authorization Expression with an active list box, text entry box, and scrollable list box appears.

For the remainder of this process, see the steps of the following procedures for creating and modifying an authorization expression.

- See “Creating an Authorization Expression for a Policy” on page 270.
- “Modifying an Authorization Expression as You Create It” on page 270.

Deleting an Authorization Expression

Before you can create a new authorization expression for a policy domain or for one of its policies, you must delete the existing one.

To delete the authorization expression for a policy domain

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain whose authorization expression you want to delete.
3. Select Default Rules.
4. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
5. Click Modify.

   The Authorization Expression edit page appears showing the content of the existing authorization expression.
6. Click the Delete All button beneath the Authorization Expression text box.

To delete the authorization expression for a policy

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain containing the policy whose authorization expression you want to delete.

   Access Manager > My Policy Domains > *policy_domain*

3. Select the Policies tab.
4. Select the name of the policy whose authorization expression you want to delete.
5. Select Default Rules.
6. Select the Authorization Expression tab.

   The Authorization Expression view page appears showing the existing authorization expression.
7. Click Modify.

   The Authorization Expression edit page appears showing the content of the existing authorization expression.
8. Click the Delete All button beneath the Authorization Expression list box.

Authorization Actions

For every authorization rule, you can configure both a set of actions to be taken if a user is granted access to the requested resource as a result of evaluation of the rule and a set of actions to be taken if a user is denied access to the resource. You can also configure sets of actions to be taken depending on the result of the authorization expression itself.

For both entities—rules and expressions—the definitive result of evaluation of the expression determines which actions are taken. Not all rules of an authorization expression contribute to the definitive result of the expression. The only actions taken are for the rules that led up to the definitive result of the expression. For explanation of the definitive result, see “About Evaluation of the Rules of an Expression” on page 250.

This section includes the following topics pertaining to actions:

- “About Actions For Rules and Expressions”
- “About Kinds of Actions”
- “About the Use of HTTP Header Variables and Cookies”
- “About Passing Information Using Actions”
- “Which Actions Are Returned?”
- “About Complementary Actions”
- “Setting Actions for Authorization Rules”
- “Setting Actions for Authorization Expressions”
- “About Duplicate Actions”

- “Setting the System Default Duplicate Actions Behavior”
- “Setting the Duplicate Actions Behavior for an Expression”

About Actions For Rules and Expressions

In addition to being able to define to whom the Allow Access part and the Deny Access part of a rule applies when you configure the rule, you can also specify separate sets of actions for each result of the rule.

You can configure actions for the following results of evaluation of rules and expressions:

- **Authorization Success**—For both rules and expressions
- **Authorization Failure**—For both rules and expressions
- **Authorization Inconclusive**—For expressions only

  This result occurs when evaluation of the rules of the expression for which the user qualifies produce conflicting results, or the user does not qualify for any rules of the expression.

Additional information about these conditions is provided in the following sections:

- For a description of the results of rules of an expression, see “About Authorization Rule Evaluation” on page 236.
- For a description of the results of expressions, see “About Authorization Expression Evaluation” on page 249.

About Kinds of Actions

Actions allow you to:

- Redirect the user’s browser to another URL.

  You can redirect URLs from the Access Server to an AccessGate or a WebGate.

- Pass information about the user to downstream applications in the same NetPoint policy domain or a different one.

  Using HTTP header variables or cookies, you can use actions to pass the following kinds of information:

  - User profile
  - User’s DN

- Static text strings

See “About the Use of HTTP Header Variables and Cookies” on page 278 for details about using header variables to pass information to downstream applications.

About the Use of HTTP Header Variables and Cookies

Consider the 4K size limit of the HTTP header when you use HTTP header variables and cookies to pass information to downstream applications. This HTTP header size limit includes all cookies, server variables, environment variables—that is, all of the content of the HTTP header. There is no constraint on the number of individual elements an HTTP header can contain, as long as the content does not exceed the 4K limit. When assessing the amount of available space in the HTTP header, take into account the byte size of the data used by NetPoint and other applications. For example, if NetPoint and other applications combined use 1K in the HTTP header, you would have 3K for your data.

How Caching Header Variables Affects their Availability

If a header variable’s value is dynamic, the value is not available until the Access Server cache is refreshed.

The refresh frequency is set in the Policy Cache Timeout field in the Access Server Configuration/*Name of Access Server* screen. If you plan to use header variables with dynamic values, ask your NetPoint Administrator about the refresh frequency.

How Web Servers Handle Header Variables

Web servers process header variables differently. This variability affects how you must implement header variables in your applications.

Here are some examples:

- Netscape/iPlanet Web servers precede NetPoint variables with the string, HTTP:
  - If you define a variable called HTTP_CN, Netscape/iPlanet produces a variable called HTTP_HTTP_CN.
  - When you write an application that needs to read a header variable, the application must look for a variable called HTTP_HTTP_CN and not HTTP_CN.
- Microsoft IIS expects header variables to be defined with a dash, not an underscore. You would enter HTTP–CN, not HTTP_CN.

  The receiving application must read the variable as if it had an underscore. It looks for HTTP_CN, not HTTP–CN.

- The Lotus Domino Web server cannot pass NetPoint header variables.

For information about how to use header variables for various servers, refer to your Web server's documentation.

About Passing Information Using Actions

Actions can pass information about users to other applications in the same or a different NetPoint policy domain. Table 21 provides examples of how to use actions.

Table 21 Using Actions to Pass Information to Other Applications

Task	Example
Personalizing the end-user's interaction with the receiving application	You can use an action to send the user's name to a downstream application. The application could use the name to greet the user with a personalized message when the user logs in.
Passing information in a header variable	You can use a header variable: • To pass membership information • To pass information about a user for purposes of single sign-on For SSO to work, the target application must be able to use the variable.
Redirecting users to a specific URL upon failure or success of the attempt to authorize	You can use redirection to send the user to another location. For example, you can redirect the user to your portal page following authorization

Which Actions Are Returned?

Different actions are returned, depending on the result of the authorization expression and the rule or rules that were decisive in producing the definitive result. The Access Server returns the actions for the results of the definitive rules—the final definitive rule and those of the rules that led up to it. It determines the actions to return based on the following considerations:

- If the result of an authorization expression is *Deny Access*, the Authorization Failure actions for all of the definitive rules are returned.

  For example, for the following compound complex authorization expression, the user qualifies for the Deny Access conditions of Rule 5, Rule 6, and Rule

7. The Authorization Failure actions are returned for all of these rules, but no actions for Rule 3 are returned.

(R5 AND R6) AND (R3 OR R7)

- If the result of the authorization expression is *Allow Access*, the Authorization Success actions for the definitive rules are returned.

  For example, for the following compound complex authorization expression, the user qualifies for the Allow Access conditions of Rule 1, Rule 2, and Rule 4. The Access Server returns the Authorization Success actions for Rule 1, Rule 2, and Rule 4, which are the definitive rules.

  (R1 AND R2) AND (R4 OR R3)

  Because Rule 4 is the final definitive rule, the Access Server stops evaluating the expression after it. It does not evaluate Rule 3 because it has no effect on the outcome.

About Complementary Actions

You can combine the actions resulting from evaluation of two or more rules to produce a desired result. For example, the Authorization Success *actions* for Rule 1 and Rule 2 in the following expression are combined to present a personalized greeting to the user for authorized users.

Rule 1 AND Rule 2 OR Rule 3

Here is how the actions for the rules are specified:

- For Rule 1, the Authorization Success action directs the Access Server to return the user's cn in the HTTP_CN header variable.
- For Rule 2, the Authorization Success action directs the Access Server to return the text 'Hello' in the header variable HTTP_GREETING.

For example, Sonal qualifies for both rules of the compound condition of the expression. She is a member of the marketing department group and the IP address of her computer is 192.168.2.123. Because she was successfully authorized as a result of evaluation of the expression, Sonal is presented with a personalized greeting when she logs into the downstream application, the resource she requested.

Working with Authorization Actions

Following discussions provide procedures to work with authorization actions:

- "Setting Actions for Authorization Rules" on page 281
- "Setting Actions for Authorization Expressions" on page 282

Setting Actions for Authorization Rules

Use the Actions feature to define an authorization rule's actions for responding to authorization success and authorization failure results. An action returns a specific value, such as the value of an attribute.

Actions you specify correspond with access conditions in the following way:

- Authorization success actions apply to Allow Access conditions.
- Authorization failure actions apply to Deny Access conditions.

To create an action for an authorization rule

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain containing the authorization rule whose actions you want to set.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.

   Note: If you are just now creating a policy domain, you do not see any authorization rules.

4. Select the authorization rule for which you want to set actions.
5. Click Actions.
6. Click Add.
7. For each of the following conditions, configure the actions to be taken—the RedirectURL and the user information to be returned:
   - Authorization Success
   - Authorization Failure
8. Click Save.

Configuring an Authorization Action When Using Disjoint Domains

If you have disjoint domains, you need to configure an authorization scheme that enables searches for users with identical user IDs who reside in disjoint domains.

To configure an authentication scheme for disjoint domains

1. In the action that you define upon success, you need to set the following values:

   Type—HEADERVAR

   Name—HTTP_OBLIX_UID

   Return Attribute—obuniqueid

   Note: This must be done for both the default identity and access policy domains.

2. In addition you need to make the following configuration file changes:

   In the following file:

   AccessManager_install_dir/access/oblix/apps/common/bin/globalparams.lst

   change the value of whichAttrIsLogin to ObUniqueID

   Make the same change in the following file:

   COREid_install_dir/identity/oblix/apps/common/bin/globalparams.xml

Setting Actions for Authorization Expressions

You can define actions for three kinds of results of evaluation of an authorization expression: authorization success, authorization failure, and inconclusive results of the expression evaluation.

To create an action for an authorization expression

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that the authorization expression whose actions you want to set belongs to.
3. Select Default Rules.
4. Select the Authorization Expression tab.
5. Click Actions.
6. Click Add.

7. For each of the following conditions, configure the actions to be taken depending on the result of evaluation of the expression (that is, the RedirectURL to use and the user information to return):
   - Authorization Success
   - Authorization Failure
   - Authorization Inconclusive
8. Click Save.

About Actions for Inconclusive Results

An inconclusive result can be returned for an authorization expression under the following two conditions:

- The user qualified for conflicting Allow Access and Deny Access rules.
- The user did not qualify for any rules of the expression.

For information about the status codes the Access Server returns when an expression is evaluated to a result of inconclusive, see “Status Codes for an Inconclusive Result” on page 249.

About Duplicate Actions

Because an authorization rule can be reused within an authorization expression, it is possible that evaluation of each instance of the authorization rule producing the same result can cause the Access Server to return the same action more than once.

It is also possible that different rules of an expression could return the same actions. Conflict can occur when, as a result of evaluation of the expression, two or more rules contributing to the definitive result produce the same actions. (See “About Evaluation of the Rules of an Expression” on page 250 for an explanation of the definitive result.)

For example, if the action of one rule is to set the HTTP_GREETING variable text string, and the action of another rule is to set the variable to a different value, a conflict occurs if the actions of both rules are returned. Because HTTP_GREETING can be set to only one text string, the Access Server must determine which one to use.

For all cases except RedirectURLs, you can set an option that determines how the Access Server should handle duplicate actions.

Important: For RedirectURL, the Access Server always returns the last URL it encounters. You cannot override this behavior.

How Duplicate Actions Are Handled

How the Access Server handles duplicate actions is defined by a system default setting, which you can configure. However, you can override the system default behavior for the individual authorization expressions of policy domains and policies. Here are the three behaviors from which you can choose:

- **Duplicate—**If you choose this option, the Access Server appends each new value it encounters to the information it returns to the application requesting authorization for the user. (The Access Server does not check for duplicate information.) Select this option if the application expects to receive information for all instances of the action. In this case, the application must process the values of all duplicate actions returned to it. Use of this option may incur performance issues.
- **Ignore Duplicate—**If you choose this option, the Access Server removes all duplicate actions, and only the first instance of the action is returned to the application requesting authorization for the user. Each time an action value is added, the Access Server checks existing values to determine if the new action duplicates an existing one. If the Access Server finds one, it does not add the new value to those it returns to the application. In this case, any information inherent to the value of the repeated action is lost.

  Because the Access Server must check for duplicate actions, use of this option may incur performance costs
- **Override—**If you choose this option, only the value of the last instance of the action is returned. Each new value overwrites the previous one, and previous values are lost. Do not select this option if the application requesting the authorization expects the results of all duplicate actions. This option is the most efficient one.

Duplicate Actions and WebGate Restrictions

The ability to process duplicate actions applies to AccessGates only. The Access Server sends to the WebGate the actions as specified by the duplicate actions policy—whether Duplicate, Ignore Duplicate, or Override. However, the WebGate supports only a single value per header variable. Although it receives the duplicate actions, the WebGate overrides duplicates so that the last value set for the header variable is used. Values set for the same header variable by previous actions are lost.

Setting the System Default Duplicate Actions Behavior

You can specify a system default setting for how the Access Server should handle duplicate actions, if any occur. By default, the system setting applies to handling of duplicate actions resulting from evaluation of all authorization expressions under control of the Access Server. However, you can override it for an individual authorization expression.

To set the system default duplicate actions behavior for the Access Server

1. Launch the NetPoint Access System, select the Access System Console, and select Access System Configuration.
2. Select Common Information Configuration.
3. Click Duplicate Actions.
4. Select the radio button to set the duplicate action behavior: Duplicate, Ignore, or Override.
5. Click Save.
6. Restart the server for the duplicate actions policy change to take effect.

Setting the Duplicate Actions Behavior for an Expression

For each authorization expression, you can specify how you want the Access Server to handle duplicate actions if any occur as result of evaluation of the expression. By setting the authorization expression's Duplicate Actions value, you override the system default Duplicate Actions behavior.

To set the behavior for handling duplicate actions for an expression

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that the authorization expression belongs to.

3. Select Default Rules.

   A page appears listing the default rules and the authorization expression for the policy domain.

4. Select the Authorization Expression tab.

5. Click Duplicate Actions.

   The Duplicate Action Headers page appears.

6. Select the radio button for the duplicate actions behavior for the expression: Duplicate, Ignore, or Override.

Creating Custom Authorization Actions

You can specify customized actions to be performed following successful authorization of a user or failure to authorize the user. Implementing a custom action requires an authorization plug-in. When defining a customized action:

- Authorization success actions apply to Allow Access conditions.
- Authorization failure actions apply to Deny Access conditions.

Refer to the *NetPoint* 7.0 *Developer Guide* for details on creating a plug-in. For information about actions, see “Authorization Actions” on page 276.

To implement a custom action

1. Launch the NetPoint Access System, select the Access Manager, and select My Policy Domains.
2. Select the policy domain that the authorization rule belongs to.
3. Select the Authorization Rules tab.

   A page appears listing the authorization rules for the policy domain.

   If you are creating a policy domain, you do not see any configured authorization rules.

4. Select the authorization rule for which you want to set custom actions.
5. Click Custom Actions.

   You are not able to select Custom Actions unless at least one authorization plug-in has been defined.

6. Click Add.
7. Enter the information for the custom action to be taken following successful authorization of a user or failure to authorize the user.
8. Click Save.

Note: You can define multiple custom actions for Authorization Success or Authorization Failure.

Authorization Schemes for Custom Plug-Ins

You can create authorization schemes for custom plug-ins that perform authorization tasks. You must be a Master Access Administrator to create and manage authorization schemes. After you create an authorization scheme, a Delegated Access Administrator can include the scheme in an authorization rule.

About Authorization Schemes and Custom Plug-Ins

NetPoint provides a default authorization scheme called Oblix Authorization Scheme that you can use for any authorization rules you create. However, you can create custom authorization schemes that include custom plug-ins used to perform different or additional tasks from those of the default scheme. After you create a custom authorization scheme, Delegated Access Administrators can include the plug-in in an authorization rule.

Historically, NetPoint has supported writing authorization plug-ins in C. With NetPoint v6.5 and later, you can now write these plug-ins using any language supported by the Microsoft .NET framework, including C, C++, and Visual Basic. For details about managed code for authorization plug-ins, see the *NetPoint 7.0 Developer Guide*.

About Authorization Plug-Ins

A custom authorization plug-in is a shared library (.dll or .so) that the Access Server uses to make outbound calls to external business logic for determining user authorization privileges and actions.

You can write a custom plug-in for any purpose. For example, you may want to look up a user's bank balance from a mainframe application to determine authorization privileges.

In some cases, the plug-in may pass authorization actions in addition to other parameters. The types of information a custom plug-in can pass are the same as those you can configure for an authorization rule. They are:

- User profile attributes
- Configuration parameters, required or optional
- Context-specific information, such as HTTP header information

Task overview: Providing customized authorization plug-ins

1. Write the custom authorization plug-in. See the *NetPoint 7.0 Developer Guide*.

   A NetPoint developer at your organization writes the custom authorization plug-in using the authorization plug-in application programming interface (API). The authorization plug-in API enables the Access Server to call external business logic to determine if a user is authorized to access a resource.

2. A Master Access Administrator configures an authorization scheme for each custom plug-in you want to use. See “Authorization Schemes for Custom Plug-Ins” on page 287.

   The scheme specifies information about the custom plug-in such as the location of the plug-in and the parameters it takes.

3. A Delegated Access Administrator with management rights for the policy domain can include the authorization scheme in an authorization rule. The authorization rule can then be included in one or more authorization expressions for a policy domain or any of its policies.

Note: A custom plug-in for authorization must be installed on each application server you want to protect.

Working with Authorization Schemes

This section includes the following sections which describe how to create and configure an authorization scheme for custom plug-ins.

Specifying Authorization Plug-In Paths and Parameters

To create an authorization scheme, you use the Authorization Management feature of the Access System Configuration component of the NetPoint Access System Console. You enter information to be passed to the shared library in the User Parameter, Required Parameter, and Optional Parameter fields when you create a scheme. An authorization scheme includes one or more custom plug-ins.

When you specify a shared library for your plug-in, you can enter either a complete path or a relative path to the plug-in. A relative path is evaluated with regard to the Access Server's installation directory.

For example:

lib/*myplug_in*

is evaluated as

AccesServer_install_dir/access/oblix/lib/*my_plug_in*

For information describing how to create shared libraries, refer to the *NetPoint 7.0 Developer Guide*.

User Parameters

User parameters are user attributes that are passed to the shared library when the authorization scheme is invoked.

By default, the user's DN (distinguished name) and IP address are passed to the shared library. You cannot change this setting. However, you can select other attributes to help identify the user requesting the protected resource.

Required Parameters

All parameters are name-value pairs. Required parameters for a plug-in are configured by the Master Access Administrator. Parameters can be passed at the authorization scheme level or at the rule level.

If you pass the parameter name-value pair at the authorization scheme level, it cannot be overridden at the rule level.

When a Delegated Access Administrator configures an authorization rule using the plug-in, he or shemust provide values in the rule for each required parameter not supplied at the scheme level. The parameters are then passed to the plug-in at runtime.

If you do not pass a required parameter name-value pair at the scheme level, you must provide it at the rule level.

Optional Parameters for Authorization Plug-Ins

Optional parameters help to define more fully the behavior of a plug-in. Optional parameters for a plug-in are configured by the Master Access Administrator. When a Delegated Access Administrator configures an authorization rule that uses the plug-in, he or shecan choose to provide a name-value pair for each of these parameters. If optional parameters are specified, they are passed to the plug-in at runtime.

For example, suppose a user allowed to access a bank account wants to withdraw more money than exists in the account. The optional parameters may specify that this account does not include overdraft protection and it may deny the user's request.

Viewing Authorization Schemes

You may want to view the contents and definition of existing authorization schemes before you create new ones.

To view configured authorization schemes

1. Launch the NetPoint Access System and select Access System Console > Access System Configuration > Authorization Management.

   The Authorization Management: List all authorization schemes screen appears.

2. Click the link for the scheme you want to view.

   The Details for Authorization Scheme page appears with the scheme's settings.

Adding an Authorization Scheme

If the existing authorization scheme does not meet your requirements, you may want to create a new one. In this case, as described in the previous sections, custom plug-ins must be available for the new scheme. Only a Master Access Administrator can create authorization schemes.

To create an authorization scheme

1. Launch the NetPoint Access System and select Access System Console > Access System Configuration > Authorization Management.

   The Authorization Management: List all authorization schemes page appears.

2. Click Add.

   The Define a new authorization scheme page appears.

3. In the Name field, type the name of the authorization scheme.

4. In the Description field, type a brief description of the scheme.

5. For the Plugin is managed code entry, if you are developing the plug-in using managed code, select Yes.
6. In the Managed Code Name Space field, enter the name space if you are using managed code. (If not, leave this field blank.)
7. In the Shared Library field, type the full path to the plug-in file or a path relative to the Access Server's installation directory without specifying the file extension.
8. In the User Parameter field, type the LDAP attributes to be passed to the plug-in.

   To pass context-specific data such as HTTP header variables to the plug-in, see "Using Context-Specific Data in an Authorization Request" on page 293.
9. In the Required Parameter field, type the name and value of parameters the policy domain authorization rule must send to the plug-in.

   If you specify the value for a parameter here, end users cannot change the value.
10. In the Optional Parameter field, type the name and value of parameters the policy domain authorization rules may send to the plug-in.

    If you specify the value for a parameter here, end users cannot change the value.

    For the User Parameter, Required Parameter, and Optional Parameter fields, click the plus (+) or minus (-) symbols to add or delete fields.
11. Click Save.

Modifying an Authorization Scheme

A Master Access Administrator is the only one who can modify an authorization scheme.

To modify an authorization scheme

1. Launch the NetPoint Access System select Access System Console > Access System Configuration > Authorization Management.

   The Authorization Management: List all authorization schemes page appears.
2. Click the link for the scheme you want to modify.

   The Details for Authorization Scheme screen appears.
3. Click Modify.

   The Modify Authorization Scheme screen appears.
4. Modify the parameters as necessary.

5. Click Save.

Deleting an Authorization Scheme

A Master Access Administrator is the only one who can delete an authorization scheme.

To delete an authorization scheme

1. Launch the NetPoint Access System and select Access System Console > Access System Configuration > Authorization Management.

   The Authorization Management: List all authorization schemes page appears.

2. Select the scheme you want to delete.
3. Click Delete.
4. Click OK to confirm your decision.

Auditing Authorization Events

An audit rule causes event-based data to be written to the audit log file. As a Master Access Administrator, you must create a Master Audit Rule in the NetPoint System Console. As a Delegated Access Administrator, you can derive audit rules from the Master Audit Rule for your policy domains and policies, but you cannot create an alternative Master Audit Rule.

There is one audit log per Access Server. You can configure the size of the audit log file and the rotation interval per server. Depending on events, the audit log may contain some duplicate audit entries.

Information Logged on Success or Failure

Different information is written to the audit log depending on whether the user was authorized to use the requested resource.

For authorization failure, if information for a user does not exist in the directory, the Access Server denies the user access to a resource. In this case, the cn attribute is written in the log entry. No other attributes are written, because none are available. Because there is not an entry for the user, attributes such as givenname have no meaning. In this case, the user requesting access to a resource had not previously been authenticated.

About Creating a Master Audit Rule and Derived Rules

You can define audit rules for a policy domain and its policies. Any audit rules you define must be derived from a Master Audit Rule. A Master Audit Rule must be created by a Master Access Administrator. Delegated Access Administrators can derive access rules from the Master Audit Rule, but they cannot create them.

For details explaining how to create and define these audit rules for policy domains and their policies, see the following sections in the policy domain chapter:

- “Auditing User Activity for a Policy Domain” on page 140
- “About Creating a Master Audit Rule and Derived Rules” on page 293
- “Creating an Audit Rule for a Policy Domain” on page 140
- “Defining an Audit Rule for a Policy” on page 141

Using Context-Specific Data in an Authorization Request

An authorization scheme can obtain data from external sources to be used in an authorization process. This data is passed to a custom authorization plug-in. Usually, this data consists of user values that are passed to the Access Server. This process allows authorization decisions to be made dynamically, based on user input. For example, if a user goes to a form to purchase an item for $1000, this $1000 amount can be dynamically evaluated against a limit—perhaps stored in a database—to determine if the purchase is authorized.

To retrieve context-specific data for an authorization request

1. Create an authorization scheme as described in “Authorization Schemes for Custom Plug-Ins” on page 287.
2. In the User Parameter field, type:

   `RA_source$name`

   or

   RA_name

   where *source* is one of the following:

   - server
   - header
   - post
   - query

- cookie

For information about the User parameter, see “User Parameters” on page 289.

If you omit *source*, sources are searched in the order shown above. Note that the Web server source (the server parameter) takes precedence over other sources. This prevents the request data, which is under control of the user, from overriding Web server data. For example, a remote_user cookie sent from a user does not override a remote_user variable sent by the Web server. The WebGate automatically extracts the requested data from the HTTP request.

If the custom client or AccessGate is created using the Access Server SDK, it is up to the application program calling the Access Server API to collect this data.

6 Configuring Single Sign-On

NetPoint's single sign-on capability enables users to access more than one protected URL or application with a single login. Before reading this chapter you should be acquainted with the terms and concepts covered in "Protecting Resources with Policy Domains" on page 95.

This chapter covers the following topics:

Prerequisites

Before attempting to configure single sign-on, you need to have a working COREid and Access System. This includes installing and configuring your directory server, the COREid System, the Access Manager and Access Server, and at least one WebGate or Access Gate. For complete details, see the *NetPoint 7.0 Installation Guide*.

About Single Sign-On

Single sign-on gives users the ability to access more than one protected resource (Web pages and applications) with one authentication. NetPoint allows you to protect Web sites and applications by defining what resources you want to protect and providing rules for accessing the resource. The rules are for:

- **Authentication**—Authentication is the process of proving that a user is who he or she claims to be. To authenticate a user, a WebGate presents the user's browser with a request for authentication credentials in the form of a challenge. The challenge is referred to as a challenge method or authentication method.
- **Authorization**—Authorization is the process of determining if a user has a right to access a requested resource. A user may want to see data or run an application program protected by a policy. The requested resource may belong to a policy domain, or it may be covered within that domain by a specific policy that is different from the global one.

For more information on protecting access to a single resource, see "Protecting Resources with Policy Domains" on page 95.

Different Types of Single Sign-On

Single sign-on can be implemented in a variety of ways:

- **Single domain**—For example, you can set up single sign-on for a set of URLs within the domain mycompany.com.
- **Multi-domain**—For example, you can set up single sign-on for a set of URLs that reside within the domains mycompany.com and yourcompany.com.
- **Applications and third-party products**—For example, you can set up single sign-on between NetPoint and a IBM WebSphere Application Server

The first two implementations use encrypted cookies, as explained in "Single Sign-On Cookies" on page 297. For these implementations to work, end users must enable their browsers to receive cookies.

Single Sign-On Cookies

NetPoint implements single-domain and multi-domain single sign-on through an encrypted cookie called the *ObSSOCookie*. The WebGate sends the ObSSOCookie to the user's browser upon successful authentication. This cookie can then act as an authentication mechanism for other protected resources that require the same or a lower level of authentication.

When the user requests access to a browser or another resource, the request flows to the Access Server. The user is logged in, and the ObSSOCookie is set. The Access Server generates a session token with a URL that contains the ObSSOCookie. Single sign-on works when the cookie is used for subsequent authorizations in lieu of prompting the user to supply authorization credentials.

When the cookie is generated, part of the cookie is used as an *encrypted session token*. The encrypted session token contains the following information:

- The distinguished name (DN) of the authenticated user
- The level of the authentication scheme that authenticated the user

  See "Authentication Schemes" on page 152 for details
- The IP address of the client to which the cookie was issued
- The time the cookie was originally issued
- The time the cookie was last updated

  If the user has not been idle, the cookie is updated at a fixed interval to prevent the session from timing out. The update interval is one-fourth of the length of the idle session timeout parameter. See "Viewing AccessGates" on page 53 for details.

Unencrypted ObSSOCookie data includes:

- Cookie expiry time
- The domain in which the cookie is valid
- An optional flag that determines if the cookie can only be sent via SSL

Security of the ObSSOCookie

The ObSSOCookie is a secure mechanism for user authentication. When NetPoint generates the cookie, an MD-5 hash is taken of the session token. When the ObSSOCookie is used to authenticate a user, the MD-5 hash is compared with the original cookie contents to be sure no one has tampered with the cookie. MD-5 is a one-way hash, so it cannot be unencrypted. The Access Server does the comparison by hashing the session token again and comparing the output with the hash of the token already present in the cookie. If the two hashes do not match, the cookie is corrupt. The system relies on the fact that if someone tampers with the session token, the hashes will not match.

The single sign-on cookie does not contain user credentials such as username and password.

Configuring the ObSSOCookie

Configuring the ObSSOCookie is a one-time activity conducted by a NetPoint Administrator or Master Access Administrator. The cookie is encrypted using a configurable encryption key known as a *shared secret*.

- For shared secret keys used in installations of NetPoint 5.x, the RC4 encryption scheme was recommended.
- For shared secret keys used in installations of NetPoint 6.x, the RC6 encryption scheme was recommended.

NetPoint 7.0 does grandfather the ObSSOCookie *only* if the shared secret is regenerated and *not* for changes in the configuration of chiper to be used. NetPoint always tries to use the newer shared secret when decrypting the ObSSOCookie. If this is not successful, it uses the older shared secret. If this fails, NetPoint queries the Access Server to see if a new shared secret was generated. If none of the keys is successful, the user is prompted to re-authenticate.

Both WebGates and AccessGates have an update thread that activates once a minute. The shared secret is updated at that time or if WebGate fails to decrypt the cookie.

The shared secret encryption algorithm is a NetPoint-wide setting. It affects all encrypted cookies, not just the ObSSOCookie.

For single sign-on to work with pre-NetPoint 6 WebGates, you must continue to use RC4 until the WebGates are all upgraded. Otherwise, the older WebGates will not be compatible with the new WebGates, and single sign-on will not work.

Note: Oblix recommends that administrators use RC6 as the encryption algorithm. It is a much stronger algorithm than RC4.

To configure the ObSSOCookie

1. Generate a key to encrypt the ObSSOCookie from the Access System Console.

   See “Creating a Shared Secret Key” on page 328 for details.

2. Decide if you want the ObSSOCookie to be sent only via SSL.

   See “Securing the ObSSOCookie in an Authentication Scheme” on page 227 for details.

Single Domain Single Sign-On

The simplest form of single sign-on occurs within a single domain. For example, suppose within the domain mycompany.com you are hosting several restricted Web sites on several hosts. You can set up single sign-on so that users with the right privileges can access all or a subset of these restricted areas after just one authentication.

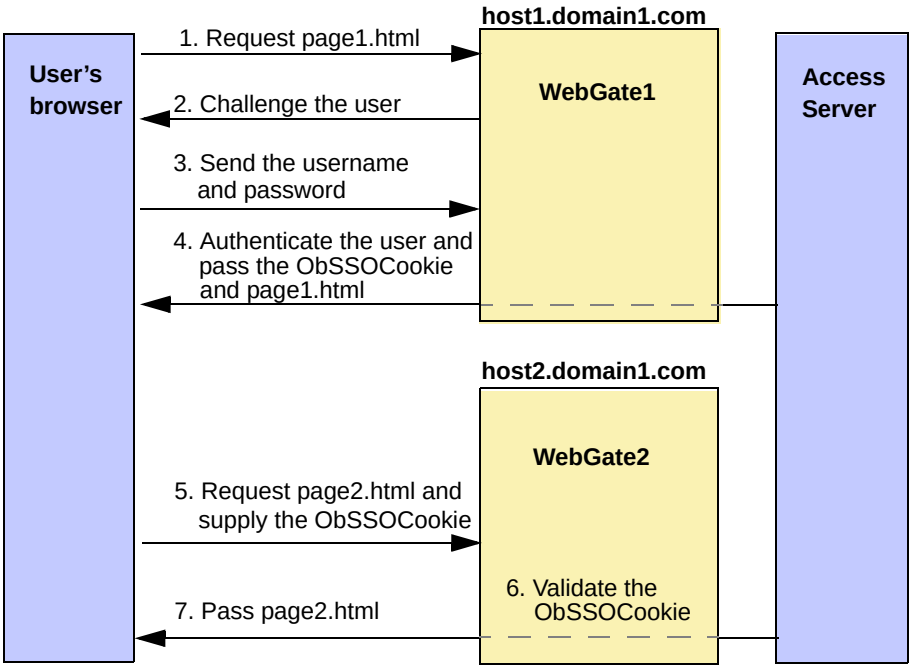
In order for single domain single sign-on to work, you need a fully functional NetPoint System, including at least two WebGates, as described in the following sections.

How Single Domain Single Sign-On Works

Single domain single sign-on works by passing the ObSSOCookie among the WebGates configured for the domain. For example, suppose a user requests index.html on Host1 through a Web browser protected by WebGate1. The process overview in Figure 11 illustrates the events as WebGate1 on Host1 asks the user for a user name and password. If the Access Server accepts the user’s authentication, the Access Server gives WebGate1 permission to give the user access to index.html. Then WebGate1 gives the user access to index.html along with the ObSSOCookie.

If this user now wants to access Host2, the user’s Web browser sends a request to WebGate2 for a page from Host2 along with the ObSSOCookie. If the two WebGates have the same cookie domain, WebGate2 can look at the ObSSOCookie and determine if the user is authenticated. The user does not have to re-authenticate.

Figure 11 Process Overview of Single-Domain Single Sign-On



In single domain single sign-on, the ObSSOCookie is associated with a particular domain, for instance, domain1.com. When the user requests resources within that domain—for example, foo.domain1.com/page2.html—the ObSSOCookie can be used to authenticate the user.

Setting up Single Domain Single Sign-On

The following is a summary of configuring a single domain for single sign-on.

Task overview: Enabling single domain single sign-on

1. Install a directory server and Web server according to the vendor's instructions.
2. Install and set up a working NetPoint system, as explained in the *NetPoint 7.0 Installation Guide*.
   a) Install and set up the COREid System.
   b) Install and set up the Access System.
3. Set up a WebGate, as described in the procedure "To configure the WebGate" on page 301.

4. Configure access controls to a resource protected by this WebGate, as described in the "Task overview: Defining authentication and authorization schemes for single sign-on" on page 302.
5. Set up a second WebGate, as described in the procedure "To configure a second WebGate for single sign-on" on page 303.
6. Configure access controls to another resource protected by the second WebGate, again using the "Task overview: Defining authentication and authorization schemes for single sign-on" on page 302.
7. Specify the same primary cookie domain for the two WebGates.

Configuring the WebGates

This discussion assumes that you have completed WebGate installation as part of your Access System installation and setup. For more information, see "Prerequisites" on page 296.

To configure the WebGate

1. From the Access System Console, click Access System Configuration > AccessGate Configuration.
2. Configure the WebGate as explained in "Adding an AccessGate" on page 60, *and* be sure to:

   a) Add a domain name for the Primary HTTP Cookie Domain.

   For example:

   host1.mycompany.com

   Note: The more general the domain name, the more inclusive your single sign-on implementation will be. For example, if you specify b.com as your primary cookie domain, users will be able to perform single sign-on for resources on b.com and on a.b.com. However, if you specify a.b.com as your primary cookie domain, users will have to re-authenticate when they request resources on b.com.

   b) Set a value for user session timeout to define how long the ObSSOCookie lasts. Use the two Access Server parameters for setting this timeout:

   Maximum User Session Time—Specifies the number of seconds that a user's connection to a resource can last before the user must re-authenticate.

   Idle Session Time—Specifies the number of seconds that a cookie can remain valid without user activity. The shorter the session, the more frequently users must re-authenticate. Shorter sessions are more secure because they leave less time for an unauthorized user to access an

unattended browser or an intercepted cookie to be re-used in a replay attack.

For more information about these parameters, see “Configuring AccessGates and Access Servers” on page 33.

3. Configure multiple ways for a user to specify the fully qualified domain name, if desired.

   For SSO to work, users must enter a fully qualified domain name. You can create alternative ways to specify the domain name, as described in “Using Host Identifiers and Host Contexts” on page 127. If a preferred host is not specified, all known variations of IP addresses and URLs must be listed in the Host Identifier. This is the only way to prevent users from typing an IP address to bypass authentication and authorization.

4. Configure access controls to another resources protected by the second WebGate, as outlined in the task overview below.

Task overview: Defining authentication and authorization schemes for single sign-on

1. Create an authentication scheme for the domain and a level for the scheme, as described in “Creating an Authentication Scheme” on page 154.

   If you use different authentication schemes on the two WebGates, users can go from a higher authentication scheme to a lower one, but not from a lower one to a higher one.

   For example, if a user is granted access to a resource that has a Basic Over LDAP authentication scheme defined as having a level of 2, the user can access other resources that have schemes with the same or a lower level. However, if the user tries to access a resource with a more stringent authentication challenge, such as a scheme called Client Certificate with a level of 5, they must re-authenticate.

2. Create an authorization scheme, as described “Adding an Authorization Scheme” on page 290.

3. Take stock of your authorization schemes and consider the following:

   Users who use single sign-on may pass the authentication tests but may fail the authorization tests when attempting to access a second or third resource. Each resource in the domain may have a unique authorization scheme.

4. Configure a second WebGate for single sign-on, as described in the next procedure.

To configure a second WebGate for single sign-on

1. Configure a second WebGate for a set of resources in the same domain.

   For example, set up a WebGate for

   host2.mycompany.com

   Note: Give the second WebGate a domain configuration identical to the first WebGate, and be sure it communicates with the same Access Server as the first WebGate.

2. From the Access System Console, click Access System Configuration > AccessGate Configuration.
3. Click the link for the *first* WebGate.
4. Click Modify.
5. In the Primary HTTP Cookie Domain field, enter the domain using a.*domain.domain* format.

   For example:

   oblix.com
6. Click Save.
7. Click Back.
8. Select the *second* WebGate, click Modify, and enter the same domain.

   Note: Be sure the primary HTTP cookie domains are *identical* for the two WebGates.
9. Save your work.

When two WebGates are set up, single sign-on should work between them. You must install a WebGate on each Web server that you want to protect.

Reverse Proxy Single Sign-On

If you are going to use a reverse proxy in a single sign-on configuration, be sure either to set the ipvalidation parameter to false or too add the proxy IP address to the IPValidationExceptions list in the WebGateStatic.lst file. You need to do this because the reverse proxy hides the client's IP address. See "Controlling Behavior with WebGateStatic.lst" on page 342 for details.

In some situations the Apache Reverse Proxy does not pass the ObSSOCookie to BEA WebLogic after a successful authentication. To avoid this issue, use Form Based authentication instead of Basic Over LDAP when using Apache Reverse Proxy with BEA WebLogic.

Logout From a Single Domain SSO Session

By default, the WebGate logs a user out when it receives a URL containing "logout." For example, logout.html or logout.pl. When the WebGate receives a URL with this string, the value of the ObSSOCookie is set to "logout."

The logout URL is configured using the WebGateStatic.lst file. WebGate treats any designated URL as a signal to log the user out of the SSO domain. If the configuration is not specified, then the default behavior is used. The default behavior assumes that any URL that contains the string *logout.* (including the ".") is a logout URL, with the exceptions of logout.gif and logout.jpg.

The following lines can be configured in the WebGateStatic.lst file:

```
# List of custom logout urls.
# Uncomment the following when the logout URL needs to be
configured.
# NOTE: For Unix systems the URL is case sensitive.

#LogOutUrls:
#BEGIN:vList
# /access/oblix/apps/common/bin/logout.html
# /logout.htm
#END:vList
```

The logout URL attribute in WebGateStatic.lst can have multiple values. For each browser request, the list of configured logout URLs is scanned to determine whether the user will be logged out of the single sign-on domain. The performance of WebGate is affected by the number of logout URLs. On Unix machines, the logout URLs are case sensitive.

Multi-Domain Single Sign-On

Multi-domain single sign-on allows a user authentication to be honored by all the hosts in two or more domains. The main objective in multi-domain single sign-on is to provide the user with an ObSSOCookie from each domain. Cookies cannot be sent across multiple domains. To achieve single sign-on across multiple domains, the Access System requires that you specify a primary domain for authentication. This primary domain acts as a central hub for all authentications. Regardless of what domain users try to authenticate to, each WebGate redirects them to the primary domain expressed as a single URL.

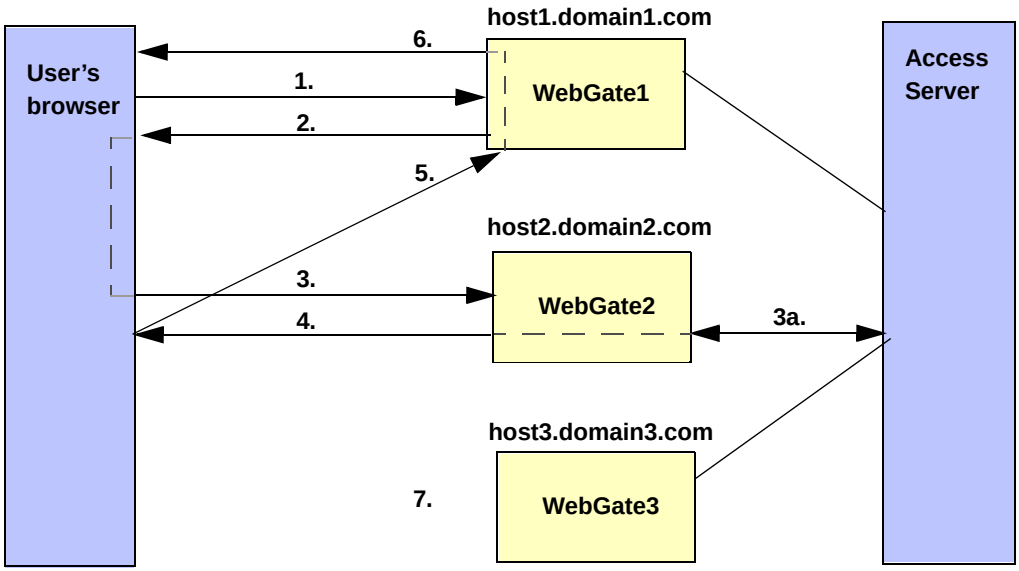
Multi-domain single sign-on is implemented, and works, in much the same way as single domain single sign-on. For more information, see “Single Domain Single Sign-On” on page 299 *and* note the following differences:

- For single domain single sign-on, you configure WebGates in one domain. However for multi-domain single sign-on, you configure WebGates on each authentication server in each domain and designate one of the authentication servers to be the primary authentication server.
- For single domain single sign-on, the WebGate provides the user with an ObSSOCookie from one domain, and that cookie is valid for each protected resource in the domain. However, for multi-domain single sign-on, a series of redirects provides the user with a different ObSSOCookie from a designated WebGate in each domain.
- For multi-domain single sign-on to work, WebGates in all domains must have access to the complete set of authentication schemes. This means that the Access Servers in your environment must use the same policy directory. If necessary, this directory can be replicated.
- Multi-domain single sign-on works only with WebGates, not AccessGates. For example, the applications discussed in “Application Single Sign-On” on page 309 have their own single sign-on methods. To integrate a scheme for AccessGate-based single sign-on with a scheme for WebGate-based multi-domain single sign on, you need to configure a proxy to act as a front end for these AccessGates.

Note: You cannot use preferred hosts with multi-domain single sign-on. An alternative is to use Challenge Redirection in your authentication schemes with form-based authentication. The challenge parameter passthrough:no enables you to redirect for authentication while delivering the user’s request to the target host and URL.

Figure 12 illustrates the process of providing the user with an ObSSOCookie from more than one domain. An explanation of the diagram follows.

Figure 12 Process Overview of Multi-Domain Single Sign-On



Process overview: Multi-domain single sign-on

1. The user initiates a request for a Web page from a browser.

   For instance, the request could be for host1.domain1.com/page1.html.

2. WebGate1 on host1.domain1.com sends the authentication request back through the user's browser in search of the primary authentication server.

   In this example, you have designated host2.domain2.com to be the primary authentication server.

3. The request for authentication is sent from the user's browser to the primary authentication server.

   This request flows to the Access Server (3a). The user logs in and the ObSSOCookie is set for domain2.com. The Access Server also generates a session token with a URL that contains the ObSSOCookie.

4. The session token and ObSSOCookie are returned to the user's browser.

5. The session token and ObSSOCookie are sent to host1.domain1.com.

6. The WebGate on host1.domain1.com sets the ObSSOCookie for its own domain (domain1.com) and satisfies the user's original request for the resource host1.domain1.com/page1.html.

7. If the user later sends a request to host3.domain3.com, a similar set of redirections takes place to set the cookie for that domain.

   Since the ObSSOCookie for the primary domain has been set, the user would not have to log in to domain3.

As mentioned earlier, implementing multi-domain single sign-on is similar to implementing single domain single sign-on.

Task overview: Implementing multi-domain single sign-on

1. Use the "Task overview: Enabling single domain single sign-on" on page 300 as a guide *and* be sure to implement the differences described in "Multi-Domain Single Sign-On" on page 304.
2. Implement redirection as described in "Using Redirection to Enable Multi-Domain Single Sign-On" on page 307.
3. Test your implementation, as described in "Testing Multi-Domain Single Sign-On" on page 308.
4. Configure logout, as described in "Logout from a Multi-Domain Single Sign-On Session" on page 308.

Using Redirection to Enable Multi-Domain Single Sign-On

For each WebGate in a multi-domain SSO configuration, you need to define an authentication scheme with redirection rules. For instance, suppose you have three authentication servers, each in a separate domain:

- host1.domain1.com
- host2.domain2.com—your primary authentication server
- host3.domain3.com

Each WebGate can only set the ObSSOCookie for its own domain. As a result, you need to create redirection rules so that when a user logs in, they are redirected to the primary authentication server. In this example, the primary authentication server is host2.domain2.com.

Note: A redirection rule is needed even for the primary authentication server.

For more information, see the next procedure.

To configure redirection

1. From the Access Server Console, click Access Server Configuration > Authentication Management.
2. Click a link for an authentication scheme.

3. In the Challenge Redirect field, enter the primary authentication server for your multi-domain single sign-on scheme.
4. Repeat these steps for WebGates in each domain in your multi-domain SSO scheme.

The steps above redirect the servers across domains to the primary authentication server.

Next, you need to be sure that the ObSSOCookie can be passed among WebGates within a particular domain.

5. Within each individual domain, ensure that each WebGate is configured to use the same primary HTTP cookie domain. See "Configuring the WebGates" on page 301.

Note: If you do not specify a primary cookie domain within a single domain, the multi-domain ObSSOCookie will not be usable by other WebGates within an individual domain.

---

6. Test your multi-domain single sign-on as described in "Testing Multi-Domain Single Sign-On" on page 308.

Testing Multi-Domain Single Sign-On

To test a multi-domain single sign-on configuration, set your browser to notify you when you receive cookies. If single sign-on is working, you should receive notification of session cookies from each domain you have configured.

Logout from a Multi-Domain Single Sign-On Session

When you log out of an application, NetPoint only removes the ObSSOCookie for the current domain. For example, if you are logged into domain1, domain2, and domain3, and you log out from domain1, only the ObSSOCookie for domain1 is removed.

The timeout of the cookie is always determined by the machine that performs the authentication. For example, suppose www.a.com sets a cookie expiration of one hour, and www.b.com sets a cookie expiration of 30 minutes. A user goes to www.b.com and is redirected to www.a.com for authentication. After 30 minutes the www.b.com cookie expires and the user is redirected to www.a.com. The cookie for www.a.com is still valid, so the user is not prompted to re-authenticate. The domain www.b.com sets a new cookie with a fresh timeout value.

You can set the www.a.com timeout value to be less than that of any other domain. This guarantees that authentication happens any time one of the other domain's cookies expires. The drawback is that if you set www.a.com expiration to be too short, single sign-on may not happen because www.a.com's cookie can expire before the user's next attempt at single sign-on. You need to determine the balance between single sign-on functionality and expiration policy.

Important: If you configure multi-domain single sign-on for your users, be sure to tell them to close all browser windows or to explicitly log out of each domain to which they are still logged in.

Application Single Sign-On

NetPoint allows you to create a web of trust in which a user's credentials are verified once and are provided to each application the user runs. Using these credentials, the application does not need to re-authenticate the user with its own mechanism. Application single sign-on allows users who have been authenticated by NetPoint to access applications without being re-authenticated.

There are two ways to send a user's credentials:

- **Using Cookies**—A specific value is set on the browser's cookie that the application must extract to identify a user.
- **Using Header Variables**—An attribute name-value pair is appended to the URL that calls the application.

With both forms of single sign-on, additional programming is required.

Header variables can be redirected only to Web servers known or protected by NetPoint. Header variables passed as authentication actions are not persistent during a user session. See "Authentication Actions" on page 211 for information about authentication actions.

For example, when a user authenticates, they may be redirected to a portal index page:

http://mycompany.com/authnsuccess.htm

For authentication failure, an authentication action may redirect the user to an error page or a self-registration script:

http://mycompany.com/authnfail.htm

For more information on application single sign-on, see the following resources:

NetPoint ReadyRealm for BEA—This is an AccessGate that is also a BEA WebLogic Custom Security Realm. It establishes a native connection between the BEA WebLogic Server and Oblix NetPoint, providing a way for WebLogic customers to use NetPoint to control user access and manage identities for their business applications. NetPoint ReadyRealm for BEA enables you to use authentication, authorization, access control, single sign-on, delegated administration, dynamic groups, workflows, identity management, and other NetPoint features to support your Web servers, application servers, and legacy platforms. See the *NetPoint 7.0 Integration Guide* for more details.

Note: To implement SSO with WebLogic Server 5.x, 6.x, and 7.x (in backward compatibility mode) in a NetPoint environment, the WebLogic Server needs to be configured to use the NetPoint BEA Realm. The Login page (either .jsp file or a servlet) must be modified as illustrated in the "Single Sign-On (SSO) with Form Login" section of the chapter "Integrating NetPoint Ready Realm for BEA" in the *NetPoint 7.0 Integration Guide*.

NetPoint Connector for WebSphere (NPCWS)—Enables applications running on IBM WebSphere to be integrated with NetPoint access control and identity management features. The NetPoint Connector for WebSphere enables J2EE resources and applications on WebSphere to use the Access System for authentication, authorization, auditing, and single sign-on. It also provides the COREid System for identity management features such as delegated administration, dynamic groups, and workflows. See the *NetPoint 7.0 Integration Guide* for more details.

The integration between NetPoint and the Plumtree Corporate Portal—Provides companies with a Web enterprise solution for building customized, secure business portals with integrated, identity-based Web access management. In this solution, the Plumtree Corporate Portal acts as a gateway to an enterprise intranet or extranet, providing users centralized access to applications and content hosted by the enterprise. See the *NetPoint 7.0 Integration Guide* for more details.

Integration between NetPoint and mySAP—Enables NetPoint single sign-on for mySAP applications and other NetPoint-protected enterprise resources and applications. It also enables you to configure NetPoint authentication schemes for mySAP applications. See the *NetPoint 7.0 Integration Guide* for more details.

Integration between NetPoint and Oracle 9iAS—Allows enables NetPoint single sign-on and identity management functionality across applications running on Oracle 9iAS, such as Oracle eBusiness Suite. See the *NetPoint 7.0 Integration Guide* for more details.

Integration between NetPoint and RSA SecurID—SecurID is a two-factor authentication product from RSA Security. NetPoint provides a plug-in and other components to provide native SecurID authentication. See the *NetPoint 7.0 Integration Guide* for more details.

Logging Out From an Application SSO Session

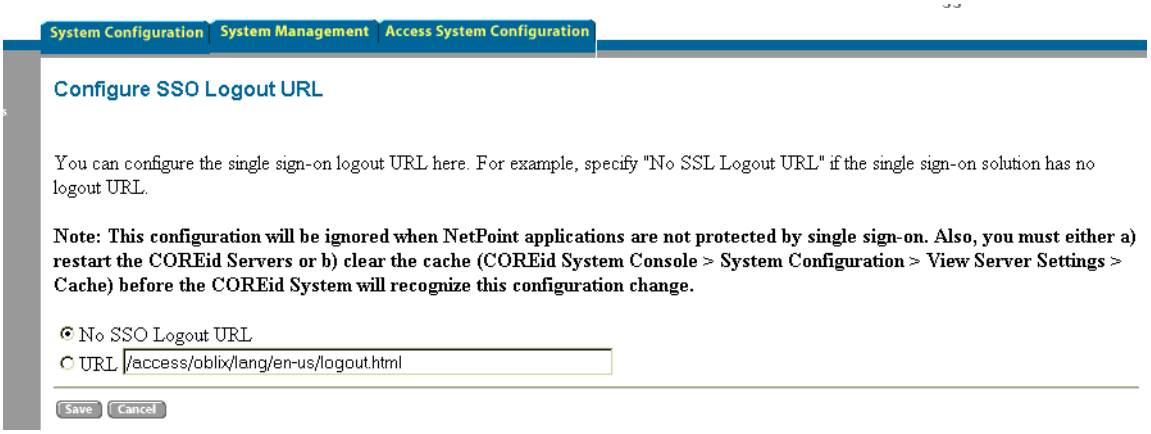
The Access System sets the ObSSOCookie for each user or application that accesses a resource protected by the Access System. The ObSSOCookie enables users to access other resources protected by the Access System that have the same or a lower authentication level. Calling the SSO Logout URL removes the ObSSOCookie, requiring the user to re-authenticate the next time they access a resource protected by the Access System.

Note: The logout.html form also contains javascript for removing the ObTemCookie set for the COREid System. It does not however, remove any cookies set by third-party applications. To ensure that users must re-authenticate, you may need to customize the single sign-on logout.html to remove these cookies.

To configure the SSO Logout URL

1. From Access System Console, click System Configuration.
2. Click View Server Settings on the side navigation bar.
3. Click Configure SSO Logout URL.

   The following screen appears.



4. Determine the logout response you prefer.
   - If you use a third-party program for logging users out, select No SSO Logout URL.
   - If you want the NetPoint COREid System and Access System to call this page automatically when the user clicks Logout, select URL.

   Note: You must manually create a link to this logout.html page from other resources protected by the Access System.

   ---

5. Click Save.

6. Flush the COREid server cache after changing the SSO Logout value.

   For more information about managing COREid Server caches, see *Volume 1* of this guide and the *NetPoint 7.0 Deployment Guide*.

SSO Between NetPoint COREid and Access Systems

You can protect the NetPoint COREid System with the NetPoint Access System just as you would any other resource.

When installing the NetPoint Access System, you can indicate that you want to protect the NetPoint applications with the Access System. This automatically creates two policy domains:

- A policy domain protecting the Access System applications starting with /access
- A policy domain protecting the COREid System applications starting with /identity

See the *NetPoint 7.0 Installation Guide* for more information.

Configuring Policy Domains for NetPoint SSO

While NetPoint provides the option to configure policy domains automatically to protect its applications during installation of the Access System, you can manually configure these policy domains at any time using the following guidelines.

Note: These guidelines assume you are familiar with the process for creating policy domains. See “Protecting Resources with Policy Domains” on page 95 for more information.

To create a policy domain that protects the NetPoint COREid applications

1. From the Access Manager, create a new policy domain as described in "Protecting Resources with Policy Domains" on page 95.
2. From the Resources tab, enter http as the resource type and enter /identity as the URL prefix.
3. From the Default Rules tab, create an authentication rule that protects the COREid applications using the challenge method of choice.

   The NetPoint Basic Authentication scheme includes the ability not to allow deactivated users access to the COREid System.
4. From the Default Rules tab, create an authorization rule that controls user access.

   Use the following screen as a guideline for configuring the authorization rule.

5. Next create the policies that allow access to key NetPoint COREid functionality such as Lost Password Management and Self Registration.

   The following four screens show a summary of the policies.

   Note: For each policy, configure the NetPoint None Authentication scheme and configure users who are allowed or denied access.

To create a policy domain that protects the NetPoint Access applications

1. From the Access Manager, create a new Policy Domain.
2. From the Resources tab, enter http as the resource type and enter /access as the URL prefix.
3. From the Default Rules tab, create an authentication rule that protects the COREid applications using the Challenge Method of choice.
4. From the Default Rules tab, create an authorization rule that allows/denies access to the appropriate users.

5. Add the same action as shown in step 4 from the previous section (“To create a policy domain that protects the NetPoint COREid applications” on page 313).
6. Create the policies that allow access to common NetPoint javascripts, gifs, and so on.
7. Configure the NetPoint None Authentication scheme and configure users who are allowed or denied access.

Displaying the Employee Type in the Top Navigation Bar

If SSO is enabled on the COREid System for connecting with another system (such as the Access System), you can use actions to define the user type in the header variables. COREid picks up this user type and displays it if there is a correct corresponding value in the obnavigation.xml file. If no user type is set, NetPoint uses the default defined in the obnavigation.xml file.

Troubleshooting SSO Between COREid and Access Systems

If you log in from one system (for example, the NetPoint Access System), and receive a message similar to the following:

> The COREid Server logged you in but the Access System (Access Manager or System Console) logged you out.

This may be due to the following:

- COREid and Access Servers are running on different machines, and the clocks are set to a different time. Solution is to synch the clocks.
- You have protected the NetPoint COREid System in a policy domain, but not the Access System, or visa versa. They must both be protected.

- The loginslack parameter.

  This parameter is located in the oblixbaseparams.lst file in *AccessManager_install_dir*/access/oblix/apps/common/bin

  This parameter affects the WebPass, which controls single sign-on between the Access Manager and the COREid System. It does not affect the single sign-on provided by WebGate. This parameter is useful only if the WebGate is not protecting the Access Manager or WebPass, and WebGate is not being used for single sign-on. In this type of scenario, the Access Manager and WebPass use a cookie for login and single sign-on which is different from the ObSSOCookie. This cookie has time stamp. The loginslack parameter controls the time difference that is tolerated between the Access Manager machine and the CoreID machine.

  The default value is 60 seconds.

Single Sign-On for Lotus Domino

By setting the remote_user header to the name of the authorized user using standard actions, you can create Domino impersonation that is similar to Windows impersonation on IIS.

Domino uses its own user store. To provide single sign-on between NetPoint and Domino, NetPoint passes a header variable, remote_user, that contains the name of the user as it is contained in the Domino user store. NetPoint looks up the user in the Domino user store, using both the long and short name stored there, and uses the preferred name defined by the Domino instance in the remote_user header.

Note: On Lotus Domino v6, be sure that the Anonymous authentication radio button on the server/ports/internet ports/web page tab is *disabled*.

To configure single sign-on using a Lotus Domino Web server

1. Create an authorization rule, as described in “Configuring User Authorization” on page 229.
2. In the General screen displaying the authorization rule, click Actions.

   The Actions page appears.
3. Click Add.
4. Under Authorization Success:

   a) Type headervar in the first Type field.

   b) Type remote_user in the Name field.

c) In the Return Value field, type the name of any attribute that identifies the user.

5. Click Save to save your changes (or click Cancel to exit the page without saving).

Enabling Impersonation in NetPoint

In a Windows environment, all processes and threads execute in a security context. Impersonation is the ability of a thread to execute in a security context that is different from that of the process that owns the thread.

When running in a client's security context, a service becomes the client to an extent. One of the service's threads uses an access token (a protected object that represents the client's credentials) to obtain access to objects for the client.

The primary purpose of impersonation is to trigger access checks against a client's identity. NetPoint overrides impersonation enabled with IIS. For details about enabling impersonation, see "Enabling Impersonation with NetPoint" on page 381.

Troubleshooting Single Sign-On

If a user experiences an problem with single sign-on, check to see if one of the following issues is the cause:

- The user's session may be timed out.

  If users are unable to maintain a session, check the value of the session timeout parameters.

- Single sign-on does not appear to work at all.

  Check the definitions for the ObSSOCookie. The cookie definition may contain the wrong domain name.

  Be sure that both WebGates have the same primary HTTP cookie domain.

  Be sure the two WebGates are in the same NetPoint installation, so that they are using the same shared secret.

- The user's authentication fails.

  Be sure this user's identity matches the authentication rules specified for the domain.

  Also be sure the user supplied a fully qualified domain name. You can configure multiple ways for a user to specify the fully qualified domain name. See "Using Host Identifiers and Host Contexts" on page 127 for details.

  Finally, verify that, on the authentication schemes to enable multi-domain single sign-on, Challenge Redirect is set.

- Users are authenticated initially, but their authorization fails when they access a resource on a second host.

  The authentication rule configured for the second host could deny the requester access to the resource. A user can go from a higher scheme to a lower scheme, but not from a lower one to a higher one. For example, if a user is authenticated to access a resource that requires a Basic Over LDAP authentication scheme, that user can access other resources having the same or a less stringent scheme. However, if the same user tries to access a resource with a more stringent authentication challenge, such as Client Certificate, the user must re-authenticate.

- The system clocks on Host 1 and Host 2 are not synchronized, and the cookie is in the future or in the past.

  If the system clocks are not synchronized, the session timeout rule may be triggered even though in reality there is no timeout issue.

- After authenticating to a protected Web site, SSO does not work when accessing a second site that has the same authentication level.

  The shared secret may have been corrupted. Regenerate the shared secret and restart the Web servers and Access Servers.

Section III: Managing the Access System

7 Access System Configuration and Management

This chapter discusses several additional Access System configuration and management functions available within the Access System Console. Topics include:

For more information about managing the Access System, see:

Prerequisites

NetPoint 7.0 should be installed and setup, as described in the *NetPoint 7.0 Installation Guide*. Read the *Introduction to NetPoint 7.0 Guide*, which provides an overview of NetPoint not found in other manuals. Also, familiarize yourself with *Volume 1*, which provides a brief review of Access System applications and installation; introduces Access System configuration and administration; and includes common functions, configuration, and administration.

About Access System Configuration and Management

Earlier chapters in this volume describe configuring administrators and viewing server settings through the Access System Console, System Configuration functions. That information is not repeated here.

Access System Configuration

Numerous functions are available in the Access System Console, Access System Configuration tab, as listed below. Unless indicated, other chapters in this volume describe Access System Configuration functions:

- **Access Server Clusters**—View existing Access Server Clusters, add new and modify existing Access Server Clusters, configure and delete Access Server Clusters.
- **AccessGate Configuration**—View existing AccessGates, add new and modify existing AccessGates, configure and delete AccessGates.
- **Access Server Configuration**—View existing Access Servers, add new and modify existing Access Servers, configure cache and audit settings.
- **Authentication Management**—Configure Authentication Rules.
- **Authorization Management**—Configure Authorization Rules.
- **User Access Configuration**—List revoked users, flush the user cache, as described in this chapter under “Configuring User Access” on page 325.
- **Common Information Configuration**—Generate a cryptographic key to encrypt cookies (covered here), configure a master auditing rule, manage resource type definitions, flush the Password Policy Cache (covered here), handle duplicate action headers. For more information on items covered here, see:
  - “Creating a Shared Secret Key” on page 328
  - “Flushing Password Policy Caches” on page 330
- **Host Identifiers**—Configure host identifiers.
- **Configure NetPoint BEA Ready Realm**—Configure the policies, workflows, and more to setup NetPoint BEA Ready Realm, as described in the *NetPoint Integration Guide*.

System Management

There are a number of options available in the Access System Console to perform system management operations, which are described in this chapter:

Diagnostics—Show Access Server details, including connection information, as described in “Running Diagnostics” on page 331.

Manage Reports—Create, view, modify, and execute User Access Privilege Reports, as described in “Managing User Access Privilege Reports” on page 331.

Manage Sync Records—Archive or purge Sync Records, as described in “Managing Sync Records” on page 334.

For information about diagnostics, auditing, reports, and logging, see *Volume 1*.

Configuring User Access

You use the User Access Configuration function available through the Access System Console, Access System Configuration tab, to manage revoked users and flush user data from the cache. For details, see:

- “Revoking Users” on page 325
- “Flushing Users from the Cache” on page 327

Note: You must be a Master Access Administrator or a Delegated Access Administrator with appropriate permissions to configure user access.

For more information on caches, see “Automatic Access System Cache Flush” on page 336. See also the *NetPoint 7.0 Deployment Guide*.

Revoking Users

You can create and modify a list of users who are prohibited from accessing any of your resources. This list supersedes any other policies controlling user access to your resources. Once a user has been revoked, if the user tries to refresh the browser, or go to another protected resource, they are denied access.

To create the revoked user list

1. In the Access System Console, click Access System Configuration > User Access Configuration.

   The User Access Configuration screen appears.

2. Click Revoked Users.

   The Modify User Revocation List screen appears, displaying the names of revoked users.

   If no revoked users exist, the Configure User Revocation List screen appears. If any exist, their names appear below the Revoked Users link.

3. Click Select User, then use the Selector feature (Select User button) to add or remove revoked users.

   See *Volume 1* of this guide for instructions on using the Selector feature.

4. Click Save to save your changes (or click Cancel to exit without saving).

Flushing Users from the Cache

This feature lets you delete information about certain users from the AccessGate and Access Server caches. For example, you might want to flush a user's information after that user's rights to view or modify an attribute have changed.

To flush user information from the cache

1. With any screen within the User Access Configuration feature displayed, click the Flush User Cache tab.

   For example:

   Access System Console > Access System Configuration > User Access Configuration > Flush User Cache

   The Flush all cached information for specified users screen appears.

2. Use the Selector feature (Select User button) to create a list of users whose information is flushed from all caches.

   See *Volume 1* of this guide for instructions on using the Selector feature.

   The names of persons you have selected appear in the above screen.

3. Click Flush Cache.

   You are prompted to confirm your decision. If you click OK the names are cleared from the screen, and information about these users is flushed from AccessGate and Access Server caches.

Creating a Shared Secret Key

You use the Shared Secret function available through the Access System Configuration, Common Information Configuration tab, to generate a key that encrypts single sign-on cookies sent from an AccessGate to a browser.

Note: You must be a Master Access Administrator to create a shared secret key. You should generate a cryptographic key as soon as possible after installing NetPoint, otherwise a less secure default is used.

AES is a new encryption scheme introduced in NetPoint 7.0. Only NetPoint 7.0 WebGates can encrypt/decrypt using this scheme. If you have a new installation of NetPoint 7.0, AES is the default encryption scheme. RC6 encryption is deprecated in NetPoint 7.0, and its support will be dropped in future releases.

If you have upgraded to NetPoint 7.0 from an older NetPoint version, the older encryption scheme (RC4 or RC6) will be retained. Older NetPoint WebGates may co-exist with NetPoint 7 WebGates:

- Use RC4 as the encryption scheme if you have NetPoint 5.x and NetPoint 7.x WebGates co-existing together.
- Use RC6 as the encryption scheme if you have NetPoint 6.x and NetPoint 7.x WebGates co-existing together.

You should use AES encryption only when all the WebGates and Access Servers are NetPoint 7.0.

Note: If the shared secret is generated more frequently than the session timeout, then the user may have a cookie that was encrypted using a shared secret that is more than two generations old. In this case, the cookie is rejected and the user is forced to re-authenticate.

To generate a cryptographic key

1. In the Access System Console, click Access System Configuration > Common Information Configuration.

   The Common Information Configuration screen appears.

2. Click the Shared Secret tab at the top of the screen.

   The Generate shared secret screen appears.

3. Click Modify.

   The Generate shared secret page now includes various ciphers from which to choose.

4. Select the appropriate cipher option for the shared secret (Oblix recommends using the AES cipher).
5. Click Generate Secret *only once*.

   NetPoint generates a new cryptographic key and distributes it to each WebGate on your system. The new key replaces the existing key without disrupting service to end users. Re-authentication only happens when the session times out. This process is called *grandfathering*. Clicking Generate Secret more than once can cause Clicking Generate Secret multiple times can put the shared secret key in COREid out of synch with the key in the Access Manager.

   A message informs you the operation was successful.

Changes to the Shared Secret Key

If you change the shared secret during a user session, the user does not need to re-authenticate. An AccessGate remembers both the old and new shared secrets, and the cookie can be decrypted with either. If a cookie is being decrypted with the old shared secret and the cookie is refreshed, it is encrypted with the new shared secret.

If the shared secret is changed more frequently than one-fourth the setting of the idle session timeout parameter, users may have to re-authenticate during a session. Otherwise, user are not required to re-authenticate during a session if the shared secret is changed.

Flushing Password Policy Caches

You use Flush Password Policy Cache function, available through the Access System Configuration > Common Information Configuration tab, to flush all password policies from the Access Server cache. Flushing the password policy cache removes existing password policies and adds newly configured policies.

Note: You must be a Master Access Administrator to flush password policy caches. You can also automatically update this cache. For more information about updates to the Access Server cache, see *Volume 1* of this guide.

To flush all redirect URLs

1. Click Access System Console > Access System Configuration > Common Information Configuration > Flush Password Policy Cache.
2. In the next screen, select the name of the policy you want to flush from the cache.
3. Click Flush Cache, and click OK to confirm your decision.
4. Click Flush Redirect URL, if you have configured redirect urls, and click OK.

Running Diagnostics

You use the Diagnostics on Access System Console, System Management page to run diagnostics on all the Access Servers in your NetPoint system or selected servers. For more information about diagnostics, see *Volume 1*.

To run diagnostics Access Servers

1. From the Access System Console, select System Management > Diagnostics.

   You are asked to select the Access Servers on which you would like to run diagnostics.

2. Select the option you want:

   - **All Access Servers**—Select All Access Servers, then click the Go button
   - **Specific Access**—Servers Hold down the Control key, then click the names of the servers whose details you want, then click the Go button.

Managing User Access Privilege Reports

You use the Manage Reports function on the Access System Console, System Management page to manage user access privilege reports.

Each Access Server can collect audit information about the resource requests it handles. The list of existing reports is visible from the Manage Reports page. In addition, you can perform the operations below:

- "Adding a Report" on page 331
- "Managing Reports" on page 333

For more information on auditing and reports, see *Volume 1*.

Adding a Report

You can create user access privilege reports that verify whether specific users have access to specific resources at specific times. Explanations to help you complete these fields appear in the procedure below.

To add a user access privilege report

1. From the Access System Console, select System Management > Manage Reports.
2. On the Manage User Access Privilege Reports page, click the Add button.
3. Complete the information requested, as follows:

   Report Name—Choose a self-explanatory name for the audit report.

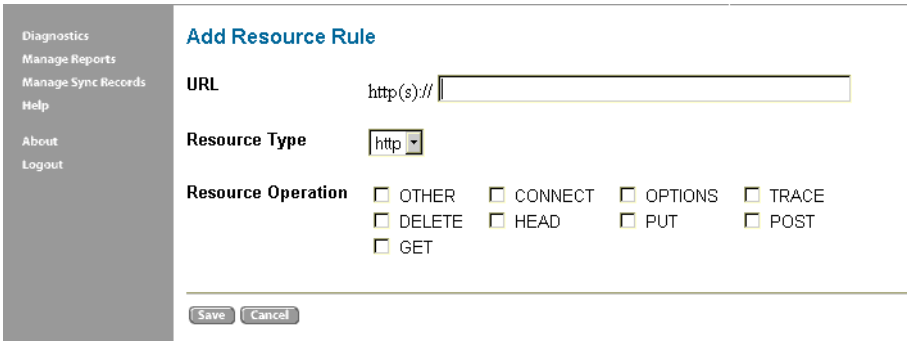
Description—If you wish, you may describe the report.

Access Server—Name of the Access server from which will collect the information for the report.

Results Storage—Indicate whether the audit data should go to a disk file or a database.

- *Store in File*—Check the box beside this option, then specify the fully-qualified path and file name in the Name of File field.
- *Store in Database*—See *Volume 1* for specific NetPoint configuration details and the *NetPoint 7.0 Installation Guide* for database support.

List of Resources—Click the Add button beside this option to display the Add Resource Rule page, as shown below.



Note: You may add multiple resources to a report. Access information on each resource will be returned in the report.

4. On the Add Resource Rule page, complete the rule by specifying the following, then click Save to return to the Add New Report page:
   - *URL*—The URL of a target resource you want to add to the report.
   - *Resource Type*—Supported choices are HTTP and EJB.
   - *Resource Operation*—Check boxes appear beside operations you can include in the report. NetPoint will determine which are permitted against the specified resources. for the specified users. at the specified time.
5. On the Add New Report page, continue specifying the information below:

   From this IP Address—*Optional*. The IP address of the machine hosting the client browser whose access you want to test. This parameter is optional.

   Date/Time of Access—Select a button to determine when a specific resource will be available to the users specified by the current report:

   - *Any*—NetPoint will determine if there is at least some point in time when the resource is available.

- *Specific date and time*—Indicates you want to identify a specific point in time so that NetPoint can determine if access is permitted at that particular moment.

Check Access for the following users—Specify whether to check the access for all users in the directory or only those you designate.

- *selected users*—Allows you to use the NetPoint Selector page to locate and add specific users. Choose *selected users*, click the Select User button, specify your search criteria, then add specific users.
- *all users*—Indicates you want to check the access of all users in the directory.

6. Click Save on the Add Reports page to save the specifications for the report and display the name you specified as a link on the Manage User Access Privilege Reports page.

Managing Reports

From the Manage User Access Privilege Reports page (Access System Console > System Management > Manage Reports), you can perform a number of operations:

- **Add**—Create a new report as described in “Adding a Report” on page 331.
- **Delete**—Check the box beside the report name on the Manage User Access Privilege Reports page, then click the Delete button to remove the report. Confirm that you want to delete the report when asked.

Note: To delete or execute multiple reports simultaneously, check all the boxes on which to operate, then click the appropriate button.

- **Execute**—Check the box beside the report name on the Manage User Access Privilege Reports page, then click the Execute button. Confirm that you want to execute the report when asked.
- **Refresh**—Update the list of reports on the Manage User Access Privilege Reports page by clicking the Refresh button.
- **Modify**—Click a link on the Manage User Access Privilege Reports page to display the Manage Existing Report page, then change the parameters for the existing static audit report. See “Adding a Report” on page 331 for details about each option.

Managing Sync Records

The Access Manager creates synchronization records, which are stored on the directory server. Over time, these records accumulate. You can manage the space these records consume on the directory server by periodically archiving or purging all the records prior to a specified date.

The archive file is typically named *nnn*.ldif, where *nnn* is a string of numbers representing both the moment at which the file was created and the *cut-off* time for archiving or purging records. All records created prior to the cut-off time will be archived or purged.

By default, the archived file is stored in:

AccessManager_install_dir\access\oblix\data\common

where *AccessManager_install_dir* represents the directory where you installed the Access Manager.

To archive sync records

1. From the Access System Console, click System Management > Manage Sync Records.
2. On the Manage Sync Records page, use the drop down lists to specify the *Date of sync records generated*.
3. Click the Archive Sync Records button.
4. When asked if you really want to archive the records, click OK to execute the action (or Cancel to revoke the operation).
5. Record the location when you are presented with a message like the one below:

```
Successfully archived 210 sync records generated before the
selected date to file /export/home/NetPoint70/webcomponent/
access/oblix/data/common/
syncrecords1090998000.20040729.040844.ldif.
```

To purge sync records

1. From the Access System Console, click System Management > Manage Sync Records.
2. On the Manage Sync Records page, use the drop down lists to specify the *Date of sync records generated*.
3. Click the Purge Sync Records button.
4. When asked if you really want to purge the records, click OK to execute the action (or Cancel to revoke the operation).

For more information about reports, see *Volume 1*.

8 Managing Access System Configuration Files

Some Access System administration tasks are performed outside the Access System Console. This chapter contains the following topics:

For more information about managing the Access System, see:

Prerequisites

NetPoint 7.0 should be installed and set up, as described in the *NetPoint 7.0 Installation Guide*. Read the *Introduction to NetPoint 7.0* manual, which provides an overview of NetPoint not found in other manuals. Also, familiarize yourself with *Volume 1*, which provides a brief review of Access System applications and installation; introduces Access System configuration and administration; and includes common functions, configuration, and administration.

Automatic Access System Cache Flush

The COREid System and the Access System use different user and group caches. You can implement automatic cache flushing for the Access System to ensure that the Access Server's cache is replaced with the latest information.

For more information about flushing the Access Server caches, see:

- "Flushing Users from the Cache" on page 327
- "Flushing Password Policy Caches" on page 330
- The *NetPoint 7.0 Deployment Guide* provides more information about NetPoint caches.

Synchronization of Access System Components

You can synchronize two aspects of the Access System:

- **System Clocks**—This is required.
- **Component Configurations**—You have the option of copying some or all configuration information from one Access System component to another.

For information on synchronizing the configuration of two Access System components, see the *NetPoint 7.0 Installation Guide*.

Synchronizing System Clocks

The clocks of all computers hosting NetPoint components must be synchronized. Without synchronization, users may not be able to log in to NetPoint components or log in to the System Console.

The two possible scenarios are:

- WebPass and Access Manager are installed on one machine, and COREid Server is installed on another machine.
- WebPass is installed on a machine without Access Manager, and is configured to route requests to two or more COREid Servers.

To implement synchronization

1. Specify a value for the loginslack parameter, located in each of these files:

   AccessManager_install_dir/access/oblix/apps/common/bin/oblixbaseparams.lst

   COREid_install_dir/identity/oblix/apps/common/bin/oblixbaseparams.xml

   where *AccessManager_install_dir* is the directory in which the Access Manager is installed and *COREid_install_dir* is the directory in which COREid Server is installed.

2. The value that you set specifies the acceptable maximum time difference, in seconds, between the two clocks.

For the first scenario, you must set the value for the loginslack parameter in both files to the same number. For the second scenario, you must set the value for the parameter in each identity server installation directory to the same number.

Reducing Network Traffic between Components

The WebGate-to-Access Server configuration polling reduces the traffic between both the WebGate and Access Server and the Access Server and the LDAP directory server.

Process overview: WebGate-to-Access Server configuration polling

1. When the WebGate is inactive for 60 seconds, it now reduces the frequency of polling for its configuration information. The polling frequency is determined by the parameter InactiveReconfigPeriod in directory oblix/apps/webgate/WebGateStatic.lst. The value for InactiveReconfigPeriod is specified in minutes. Within ten seconds of resuming activity, the WebGate performs reconfiguration polling once per minute.
2. At startup, the WebGate now checks the bootstrap configuration to see if any important parameters have changed. This makes the re-initialization process unnecessary in most cases and reduces the transient Access Server load.
3. WebGate and Access client configurations are now cached in the Access Server. The default cache timeout is 59 seconds. This should cause no modifications to the system behavior on non-Apache Access clients. The Apache Web server with WebGate now avoids unnecessary hits to the directory server. The caching parameters can be set in the oblix/apps/common/bin/globalparams.lst file. The parameter clientConfigCacheMaxElems sets the maximum size of the cache (default 9999). The parameter

clientConfigCacheTimeout determines the maximum lifetime of any element in the cache (default 59 seconds).

4. When requested by a WebGate, the shared secret used for cookie encryption is cached in the Access Server. This cache persists for a non-adjustable duration of ten minutes.

There are two ways to reduce off-time network traffic between both the WebGate and Access Server and the Access Server and the LDAP directory server:

- Changing WebGate polling frequency for configuration information.
- Changing the default configuration cache timeout for WebGate and Access client configurations that are cached in the Access Server.

Changing WebGate Polling Frequency

One way to reduce off-time network traffic between both the WebGate and Access Server and between the Access Server and the LDAP directory server is to change the WebGate polling frequency for configuration information.

To change the configuration polling frequency

1. Add the following parameter to the WebGateStatic.lst file located in *WebGate_install_dir*/access/oblix/apps/webgate.

   `InactiveReconfigPeriod`

   WebGate_install_dir is the directory where WebGate is installed.

2. Specify the value for InactiveReconfigPeriod in minutes.

   The default is 1 minute. When the WebGate is inactive for more than 60 seconds (for example, when no authentication requests are being processed), it reduces the frequency of polling for its configuration information. Within ten seconds of resuming activity, the WebGate resumes reconfiguration polling once every minute.

   See also “Controlling Behavior with WebGateStatic.lst” on page 342.

Changing Default Configuration Cache Timeout

A second way to reduce off-time network traffic between both the WebGate and Access Server and between the Access Server and the LDAP directory server is to change the default configuration cache timeout for WebGate and Access client configurations that are cached in the Access Server.

To change the default configuration cache timeout

1. Navigate to the globalparams.lst file located in:

   WebGate_install_dir/access/oblix/apps/common/bin/globalparams.lst

   where *WebGate_install_dir* is the directory where WebGate is installed.

2. Add the following parameters and specify their values:

   - clientConfigCacheMaxElems

     The default value is 9999.

   - clientConfigCacheTimeout

     The default value is 59 seconds.

   The default values listed should cause no change in the system behavior on non-Apache Access clients. An Apache Web server with WebGate will now avoid excessive hits to the directory server.

Reducing Overhead for Viewing Policy Domains

You can reduce overhead on the My Policy Domains page by turning off the display of the Resource Type and URL Prefix columns on that page. Note that these columns may contain useful information, so the gain in performance is a tradeoff.

To turn off the display of Resource Type and URL Prefix columns

1. Locate the *AccessManager_install_dir*/access/oblix/apps/common/bin/globalparams.lst file.

   where *AccessManager_install_dir* is the directory where Access Manager is installed.

2. Set the value of the parameter limitAMPolicyDomainResourceDisplay to true.

   By default, the value of this parameter is false. The Resource Type and URL Prefix columns are displayed by default. For more information on Policy Domains, see “About Policy Domains and Their Policies” on page 101.

Customizing the Access Manager User Interface

When you invoke the Access Manager, the My Policy Domains page is displayed. This page lists all of your policy domains. If you are interested in a certain policy domain, you can scroll through the list to find it. If you are responsible for a large number of policy domains, the list will be long. An easier and faster way to find the desired policy domain would be to search for it by name.

Rather than displaying the My Policy Domains page as the first page you see in the Access Manager, you may set the Search page as the default. In addition, you may customize the Search page. Topics here explain:

- "Setting the Search page as the Default Page" on page 340
- "Customizing the Access Manager Search Interface" on page 341

For additional information on customizing NetPoint, see the *NetPoint 7.0 Customization Guide*.

Setting the Search page as the Default Page

With the NetPoint Access System, you can change the first page displayed by the Access Manager from the My Policy Domains page to the Search page. The NetPoint Administrator responsible for the Web server can change the default by modifying the Oblix base parameter list file, oblixbaseparams.lst. Changes made to this file occur at the Access Server level. If you change the default, it affects all users of the Access Manager.

To set Search as the default page

1. Open the following file in an editor:

   AccessManager_install_dir/access/oblix/apps/common/bin
   oblixbaseparams.lst

   where *AccessManager_install_dir* is the directory where Access Manager is installed.

2. Locate the following section in the file:

   policyservcenter_application_info:

3. Change the entry below as follows:

   From—
   PROGRAM:../../policyservcenter/bin/policyservcenter.cgi

   To—
   PROGRAM:../../policyservcenter/bin/
   policyservcenter.cgi?program=navbar&selected_prog=
   searchframepage

4. Save the file and close it.
5. Restart the Web server.

Customizing the Access Manager Search Interface

When you perform a search in the Access Manager, the default number of results shown is 8. This means that 8 results are displayed just below the search bar. You may want to change the default value. You may also want to limit the type of searches by altering what appears in the Access Manager Search page drop-down list, which by default includes the values below:

- *That Contains*
- *Contains in Order*
- *That Begins with*
- *That Ends with*

For more information, see the following procedures:

- "To change the default number of search results" on page 341
- "To change search parameters" on page 341

To change the default number of search results

1. Locate and open the file below in a text editor:

   AccessManager_install_dir\access\oblix\apps\common\bin\oblixbaseparams.lst

2. Change the default value of defaultDisplayResultVal to a number other than 8.
3. Save the file, and restart the Web server.

To change search parameters

1. Locate and open in a text editor the policyservcenparams.lst file:

   AccessManager_install_dir\access\oblix\config\policyservcenparams.lst

2. Locate the ObEnhanceSearchList parameter and values below:

```
\ObEnhanceSearchList:
BEGIN:vNameList
OOS:MOOS
OSM:MOSM
OBW:MOBW
OEW:MOEW
END:vNameList
```

3. Comment out, or delete, the values from the list of four values above.
4. Save the file and restart the Web server.

Controlling Behavior with WebGateStatic.lst

Certain aspects of Access System behavior are controlled by the WebGateStatic.lst file. A sample WebGateStatic.lst file is shown below. Lines that are preceded with a pound sign are commented out:

```
BEGIN:vCompoundList
DenyOnNotProtected:false
CachePragmaHeader:no-cache
CacheControlHeader:no-cache
IPValidation:false
# Set UseIISBuiltinAuthentication to true
# if you are using MSPassport or Integrated
# Windows Authentication on this machine.
# Otherwise leave it set to false
# Only used for IIS
UseIISBuiltinAuthentication:false
#IPValidationExceptions:
#BEGIN:vList
#10.10.50.101
#10.10.50.102
#END:vList
WaitForFailover:-1
#InactiveReconfigPeriod:5
END:vCompoundList
```

Values in this file are described in the following paragraphs.

DenyOnNotProtected—The default value for this parameter is false. In this case, there is no protection, and access is enabled. More importantly, access may be granted inadvertently. For example, if someone attempts to access a resource using the decimal value of an IP address in the URL, access may be granted unless the host identifier includes this representation of the address.

Important: Setting DenyOnNotProtected to true is the most secure way to protect Web server content.

If you set DenyOnNotProtected to true, all requests for Web pages on the Web server protected by the WebGate are denied unless access is explicitly allowed by a policy. When this is set to true, you need to create an anonymous authentication method and allow access to content using an anonymous access policy. For information describing how to use the DenyOnNotProtected switch, see "Configuring AccessGates and Access Servers" on page 33.

CachePragmaHeader and CacheControlHeader—By default, these two parameters are set to no-cache. This prevents WebGate from caching data at the Web server application and the user's browser. However, this may prevent certain operations such as downloading PDF files or saving report files when the site is protected by a WebGate. You can set the WebGate caches to different levels. See http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html section 14.9 for details. All of the cache-response-directives are allowed. For example, you may need to set both cache values to public to allow PDF files to be downloaded.

IPValidation and IPValidationExceptions—IPValidation can be set to true or false to turn on and off IP address checking for the end user. The default IPValidation setting is true. WebGate uses the IP address in the ObSSOCookie for single sign-on. This can cause problems with certain Web applications. For example, Web applications managed by a proxy server typically change the user's IP address, substituting the IP address of the proxy. This prevents single sign-on using the ObSSOCookie. To configure single sign-on between WebGate and an access client that does not have the client IP address at authentication, the IP validation has to be explicitly turned off. To do this, you set IPValidation=false. When the IPValidation parameter is set to false, the browser or client IP address is not used as a part of the ObSSOCookie.

The IPValidationExceptions parameter syntax is:

```
IPValidation:true
IPValidationExceptions:
BEGIN:vList
10.10.50.102
10.10.50.101
END:vList
```

The above IP addresses are examples. You can add as many IP addresses as needed. These addresses are the actual IP addresses of the client, not the IP addresses that are stored in the obSSOCookie. If a cookie arrives from one of the exception IP addresses, NetPoint ignores the address stored in the ObSSOCookie cookie for validation. For example, the IP addresses in the IPValidationExceptions parameter can be used when the IP address in the cookie is for a reverse proxy.

UseIISBuiltinAuthentication—By default, this parameter value is false. Set UseIISBuiltinAuthentication to true only if you are using Microsoft Passport or Integrated Windows Authentication on this machine. It is used only for IIS, and is ignored if the WebGate is installed for another type of Web server.

WaitForFailover—In NetPoint 6.x, the WaitForFailover parameter has been replaced with the *Access Server Timeout Threshold* parameter that can be accessed through the NetPoint GUI (Access System Console > Access System Configuration > AccessGate Configuration). The WaitForFailover parameter is used only for backward compatibility with NetPoint 5.x.

Both the WaitForFailover parameter, and its replacement the *Access Server Timeout Threshold* parameter, control the TCP/IP timeout between the WebGate and the Access Servers it communicates with. The default value is "-1," which means the network default TCP/IP timeout value is used.

Note: Be sure that both the WaitForFailover parameter in the webgatestatic.lst file, and its replacement, the Access Server Timeout Threshold parameter, use the same value.

Assume a WebGate is configured to talk to one primary Access Server and one secondary Access Server. If the network wire is pulled from the primary Access Server, the WebGate waits for the TCP/IP timeout to learn that there is no connection to the primary Access Server. The WebGate tries to re-establish the connections to available servers starting with the primary Access Server. Again, the WebGate waits for the TCP/IP timeout to determine if a connection can be established. If it cannot, the next server in the list is tried. If a connection can be established to another Access Server (either a primary or secondary), the requests are re-routed. However this can take longer than desired.

Rather than rely on the default TCP/IP timeout, you can specify the Access Server Timeout Threshold in the Access System Console > Access System Configuration > AccessGate Configuration. The default value of -1 means the default network TCP/IP timeout is used. A typical value for this parameter is between 30 and 60 seconds.

When finding new connections, WebGate checks the list of available servers in the order specified in its configuration. If there is only one primary Access Server and one secondary Access Server specified, and the connection to the primary Access Server times out, the WebGate still tries the primary Access Server first. As a result, the WebGate is unable to send requests to an Access Server for a period greater than twice the setting in the WaitForFailover period.

InactiveReconfigPeriod—The WebGate has an update thread that reads the shared secret from the Access Server every 1 minute when the WebGate is active. The Access Server server returns the shared secret in its own cache (the AAA cache); this value is updated every 10 minutes. For example, the Access Server reads the shared secret from the directory at an interval of 10 minutes and this cached value is returned to WebGate.

In the idle state the WebGate reads the shared secret from the Access Server using the InactiveReconfigPeriod value. If this value is not set in the Webgatestatic.lst file, the WebGate polls the Access Server for the shared secret value at an interval of 1 minute even though the updated shared secret value will be returned only after 10 minutes.

SECTION IV: APPENDICES AND INDEX

A Form-Based Authentication

Authentication involves determining what authentication method is required for a resource, gathering credentials over HTTP, and returning an HTTP response that is based on the results of credential validation.

Form-based authentication allows you to create customized Web forms that process user logins using NetPoint's authentication and authorization mechanisms. These forms are HTML pages that allow you to present login information in different languages, to display user interface elements that comply with your company's presentation standards, and to add functions to the login page—for example, for lost password management.

This chapter covers the following topics:

About Form-Based Authentication

If a Web resource is protected using a policy with an authentication scheme that requires a form, and there is no valid session cookie (ObSSOCookie) or valid session cookie from a lower authentication level (regardless of the challenge method), NetPoint challenges the user with the form configured in the authentication scheme. The authentication challenge is an HTML form with one or more text input fields for user credentials.

In a typical form-based authentication, text boxes are provided for the user name and password. Users enter their credentials in these fields. The most common credential choices are user name and password, but any user attributes can be used, for example, user name, password, and domain. A Submit button posts the content of the form. When the user clicks the Submit button, the form data is posted to the Web server. WebGate intercepts and processes the form data. Upon validation of the user credentials collected in the form, the user is authenticated.

You may want to use form-based authentication for reasons such as the following:

- To use your organization's look and feel in the authentication process.

  For example, a custom form can include a company logo and a welcome message instead of the standard username and password pop-up window used in Basic authentication.

- To gather additional information at the time of login.

- To provide additional functionality with the login procedure, such as a link to a page for lost password management.

Note: The forms that you create for form-based authentication can be used to collect user credentials. The actual process of authentication and authorization is handled by other NetPoint functionality, described in "Protecting Resources with Policy Domains" on page 95.

The following is a summary of steps for configuring form-based authentication. For more details on this process, see "Configuring Form-Based Authentication" on page 356.

Task overview: Configuring form-based authentication

1. Create an HTML form where the user's credentials, such as user name and password, can be submitted, using information in "Considerations when Creating a Form" on page 355.

   Note: Do not protect the form or any of its components (such as GIFs and links) with an authentication method, or use a None authentication scheme.

   ---

2. Place the form in an unprotected directory, or in a directory protected by a None authentication scheme, on your Web server with a WebGate.

   The same login form and its associated authentication scheme can be used by multiple policy domains.

3. Set up an authentication scheme to use form-based authentication and define the path to the login form, as described in “Configuring a Form-Based Authentication Scheme” on page 357.

   See also “Creating Authentication Schemes” on page 153.

4. Call the form action using HTTP Get or Post, as described in “About the Form Action” on page 358.

5. Protect the target URL in the action of the login form with a policy, as described in “Configuring User Authentication” on page 149.

6. Configure the challenge parameters and passthrough mode in the authentication scheme, as described in “Challenge Parameters” on page 351.

7. Specify the plug-ins, as described in “Plug-Ins Used with Form-Based Authentication” on page 353.

Challenge Parameters

When you select the Form challenge method, you are required to provide the following three parameters in the Challenge Parameter fields.

Challenge Parameter	Description
`form:`	Indicates where the HTML form is located relative to the host’s document directory. For example, `form:/login.html`.
`creds`:	Lists all fields used for login in the HTML form. Creds: is a space-separated list. For example: `creds:login password`. **Note**: You can specify the creds parameter for the other types of challenge methods.
`action`:	The URL that the HTML form is posting to.

Note: During form based authentication with a custom plug-in, the original resource name is not available to the plug-in in the pre-defined names within the Challenge Parameter *creds* list. For example, in the Authentication Plug-in API the ObAnPluginInfo struct contains the Creds data type where the Access Server provides four pre-defined names within this list: Resource, Operation,

RequesterDN, and RequesterIP. When using form-based authentication, the Resource returned by the API is the resource that the login form POSTs to (not the actual resource of the original URL).

A fourth parameter—passthrough—is optional.

`passthrough`:	This parameter value determines whether the WebGate redirects the browser back to the original requested resource or passes the login credentials on to another program. NetPoint assumes that the URL given for the form in the authentication scheme is on the same machine as WebGate. Possible values are yes or no: Accept the default value of no if you want WebGate to redirect the browser back to the original requester resource. This omits a form challenge parameter. Specify yes if you want to pass the login credentials through to a post-processing program.

Enter passthrough:yes if you want to pass the login credentials to a post-processing system. For example, you enter passthrough:yes if you want to pass the login credentials through to a post-processing program for SSO to another application that does not accept header variables.

If you accept the default passthrough mode but want to redirect users to a page other than the originally requested resource, in the policy domain rule specify a redirection to another page upon authentication success. If redirection to the login form occurs as described in “Redirection” on page 352, and passthrough mode is not set for the form authentication scheme, WebGate redirects the browser back to the originally requested resource. You can use the ObRequestedUrl header variable to redirect.

Redirection

If the login form is the page that user requests, redirection is not needed. However, users can attempt to go around a login form, for example, by bookmarking pages. In these cases, WebGate redirects the request to the login form. After authentication success, WebGate redirects the user back to the requested resource.

A cookie named obFormLoginCookie maintains the original request information. By default, this cookie is set when the browser is first redirected to the form. Information in this cookie includes:

- The requested URL

- The requested operation
- An authentication scheme
- The host to return to URL

Without this cookie, WebGate would be unable to send the originally requested resource upon authentication.

When the user authenticates, the ObSSOCookie is also set. For more information on the ObSSOCookie, see “Single Sign-On Cookies” on page 297.

Plug-Ins Used with Form-Based Authentication

You need several plug-ins to work with your form authentication scheme. The order of the plug-ins is also important.

Credential Mapping Authentication Plug-In—Credential mapping is defined for each login form. The credential_mapping plug-in performs the task of mapping the user-supplied credentials to a unique DN in the directory server. WebGate searches the directory for profiles with attributes matching the form credentials. It handles the password credential consistently with basic authentication.

Logically, password validation can only happen after the user is identified. Therefore, the credential_mapping plug-in needs to be used before validate_password and must be the first plug-in specified in your form-based authentication scheme.

Validate Password Authentication Plug-Ins—Form authentication uses the same validate_password plug-in that is used in basic authentication. You can configure the name of the password field.

More Possible Custom Authentication Plug-Ins—As with basic authentication, custom authentication plug-ins can be used to check the username and password using other login services and user repositories. In fact, the same processing functions could be used for both basic and form username/password authentication. Custom authentication plug-ins can also process other user credential data.

Note: During form based authentication with a custom plug-in, the original resource name is not available to the plug-in in the pre-defined names within the Challenge Parameter *creds* list. For example, in the Authentication Plug-in API the ObAnPluginInfo struct contains the Creds data type where the Access Server provides four pre-defined names within this list: Resource, Operation, RequesterDN, and RequesterIP. When using form-based authentication, the Resource returned by the API is the resource that the login form POSTs to (not the actual resource of the original URL).

For more information about plug-ins, see “Configuring a Form-Based Authentication Scheme” on page 357 and the *NetPoint 7.0 Administration Guide Volume 1*.

Session Cookie and Authentication Actions

If WebGate intercepts the form login, it can build the session cookie and carry out the authentication actions.

Note: If a form authentication scheme on IIS is configured with the passthrough option, and the target of the login form requires the data posted by the form, the WebGate extension method (where the WebGate DLL is the action of the form) cannot be used. The WebGate filter method (where the action of the form is a protected URL that is not the WebGate DLL) must be used instead, and the postgate DLL must be installed and enabled. See the *NetPoint 7.0 Installation Guide* for details.

Header Variables

Form-based authentication schemes can pass authorization actions in header variables. However, they *cannot* pass *authentication* actions in header variables.

Using Context-Specific Data in an Authentication Request

An authentication scheme can collect context-specific information before submitting the request to the Access Server. Context-specific information can be in the form of an external call for information. This information can be of the following types:

- server—variables set by other Web server plug-ins
- header—HTTP header variables
- post—posted data
- query—query string data
- cookie—HTTP cookie

To retrieve context-specific data for an authentication request

1. Create an authentication scheme as described in “Creating Authentication Schemes” on page 153.
2. In the Challenge Parameter field, specify the following:

   creds:source$name

   or

   creds:name

   where source is one of the following:

   - server
   - header
   - post
   - query
   - cookie

If you omit the source, sources are searched in the order shown above.

Note: The Web server source (the server parameter) takes precedence over other sources. This prevents the request data, which is under control of the user, from overriding Web server data. For example, a remote_user cookie sent from a user will not override a remote_user variable set by the Web server.

If the client is a WebGate, as opposed to the Access Server SDK, the WebGate will extract the requested data. If the client is the Access Server SDK, it is up to the calling program to collect this data.

For a plug-in to make use of the creds parameter, you specify what is passed in the obMap credentials parameter of the ObUserSession object. See the *NetPoint 7.0 Developers Guide* for details.

Considerations when Creating a Form

You need to create a custom form that you want users to see when they access a protected resource. The form can be as complex as you want it to be. Within the form, you must at least provide fields for a user to submit a login and password.

Note: Do *not* protect the form or any of its components (such as GIFs and links) with an authentication method, or use a None authentication scheme.

Key areas to consider when you are designing a form are:

- ObFormLoginCookie as described in “ObFormLoginCookie” on page 356
- Form action, as described in “About the Form Action” on page 358
- Form action and WebGate.dll, as described in “Notes for Microsoft IIS” on page 360

ObFormLoginCookie

As previously mentioned, WebGate sets the ObFormLoginCookie when the browser is first redirected to the form. This can become a problem in the following situations:

- If your login form has a link for Password Management that is protected by a None authentication scheme, the user is redirected back to the login form instead of going to the lost password link.
- After the login has been completed, WebGate marks the ObFormLogin Cookie “done” and will not allow the user to use the form login again within the same browser instance. This causes a problem for the oblogout functionality. When a user tries to log out, and then log back in, WebGate bypasses the form login processing.

You can avoid these situations by entering an action challenge parameter when you configure your form authentication scheme. See “Protecting Resources with Policy Domains” on page 95 for more information.

Configuring Form-Based Authentication

The following procedures describe how to configure a form and an authentication scheme for the form.

Task overview: Creating a form for authentication

1. Create a custom form that you want users to see when accessing a protected resource, using considerations described in “Considerations when Creating a Form” on page 355.

   Note: Do not protect the form or any of its components (such as GIFs and links) with an authentication method, or use a None authentication scheme.

2. Place the form in an unprotected directory, or in a directory protected by a None authentication scheme, on your Web server with WebGate.

   The same login form and its associated authentication scheme can be used by multiple policy domains.

3. Configure a form-based authentication scheme, as described in “Configuring a Form-Based Authentication Scheme” on page 357.

Configuring a Form-Based Authentication Scheme

When you create an authentication scheme you include the name, an optional description, and the level of the authentication scheme. Parameters and options are described within the following procedure. For more information about authentication schemes, see “Configuring User Authentication” on page 149.

To configure a form-based authentication scheme

1. In the Access System Console, click Access System Configuration > Authentication Management > Add.

   The Define a New Authentication scheme screen appears.

2. Enter the following for the authentication scheme:

   - A name
   - A description
   - The level of the authentication scheme—The level of the scheme is a number that corresponds to the relative security level for this scheme. Higher levels are considered more secure by NetPoint.

3. Select Form as the Challenge Method, as described in “About Challenge Methods” on page 158.

4. In the Challenge Parameter field, enter the following:

   ```
   form:relative_form_URL
   creds:credential_names
   action:Action_URL
   passthrough:[yes] (Optional)
   ```

   - NetPoint assumes the *relative form URL* given for the form in the authentication scheme is on the same machine as WebGate.

     Do not include the http://*server host*:*port* portion of the URL.

     For example:

     ```
     form:/login.html
     ```

   - *Credential names* are a space-separated list of expected credential names from the form.

     For example:

     ```
     creds:login password
     ```

- The *Action URL* sets the ObFormLoginCookie to be returned only when the form posts the login credentials.

  For example:

  ```
  action:/access/dummy.cgi
  ```

  For more information, see “About the Form Action” on page 358.

- The default *passthrough* mode is no. Accept the default if you want NetPoint to automatically redirect users to their original requested resource.

5. Specify whether or not you want the user to authenticate using SSL.

   You can also use Challenge Redirect to redirect the users to a central location storing all forms.

6. If you answered yes to SSL, specify the Challenge Redirect URL for your secure server.

7. Enter the following two required plug-ins:

Order	plug-in Name	plug-in Parameters
1	credential_mapping	`obMappingBase="o=company,c=us" (the base DN in the LDAP search). obMappingFilter="[(COREid Login Attribute=%form input field for login%)]"`
2	validate_password	`ObCredentialPassword="[form input field for password]"`

Important: The *directory login attribute* is an attribute defined in COREid using a semantic login type, as discussed in *Volume 1* of this guide. Also, you cannot have spaces in the filter. The Access Manager does not validate the string that you provide as the credential_mapping filter, so it is possible to enter an erroneous filter. No error occurs while saving; however, the filter will fail and the plug-in will return “Authentication Failed” each time it is run.

For information about users and the obMappingFilter, see “Including Users in the obMappingFilter” on page 360.

8. Click Save.

About the Form Action

The form action does not process the credentials for authentication. This is the job of the NetPoint plug-ins that you configure for the form-based authentication scheme. In the *form* element of a login form, the action attribute is a URL to which form data is posted when the user submits the form.

For example, in the following form the action URL is /access/dummy and the method is post:

```
<html>
<form name="myloginform" action="/access/dummy" method="post">
UserID <input type="text" name="userid" size="20"
value="user1k1">
Password <input type="password" name="password" size="20"
value="oblix">
<input type="submit" name="submit" value="Login">
</form>
</html>
```

The action URL is configured so WebGate sets the ObFormLoginCookie for the action URL path, and this cookie is only returned on the form post. When a user submits credentials, the form action is called using the HTTP GET or POST method. The form action does not process the user's credentials for authentication. That is the job of the plug-ins configured for the form-based authentication scheme.

The form action can be a call to a URL that does not do anything. When the form posts to an action URL, WebGate intercepts the post because of the ObFormLoginCookie. WebGate processes the credentials in the post data, authenticates the user, and redirects the user to the originally requested URL as indicated by the ObFormLoginCookie. Since the action URL is never reached, it does not actually have to exist. All that is required is that a NetPoint policy protect the action URL. In the form example above, the action URL /access/dummy is protected by a policy domain that protects all URLs subordinate to /access. However, /access/dummy, as the name implies, does not exist.

The form action can also be a call to a script that does post-authentication processing. For example, you may have a script that does post-processing on credentials to achieve single sign-on for an application that does not accept header variables. When the form action is a script, the authentication scheme must be configured with the passthrough:yes challenge parameter. This tells WebGate that the action URL is a script that must be executed after the form login. In this case, WebGate does not redirect the user to the originally requested URL. WebGate allows the Web server to continue processing the action URL. WebGate passes the originally requested URL in the ObRequestedURL header variable to the action URL script, and the script can redirect to the original URL if desired.

Note: The form action URL must reside in a policy domain protected by the Access System.

Notes for Microsoft IIS

Because of the IIS architecture, the WebGate ISAPI plug-in checks all incoming requests for post-processing data. You must do *one* of the following:

- Either set your form action to call the webgate.dll, for instance:

  ```
  action="/access/oblix/apps/webgate/bin/webgate.dll"
  ```

  Note: With NetPoint 6.5, a new directory structure was instituted to accommodate localization. Before NetPoint v6.5, the form action contained a different path to webgate.dll.

- Or ensure the WebGate filter post-processing is turned on by setting the following Registry entry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Oblix\Oblix NetPoint\version\WebGate\postdata=″yes″
```

  where *version* is the version number of the installed NetPoint product.

Including Users in the obMappingFilter

This topic describes:

- "Including Only Active Users" on page 360
- "Including Non-Active Users" on page 361

Including Only Active Users

You may want to include only activated users in your obMappingFilter so that only activated users can login. To do this, you must filter users whose obuseraccountcontrol=ACTIVATED.

To include only active users in the obMappingFilter

1. Follow the procedure "To configure a form-based authentication scheme" on page 357.
2. In the mapping filter, specify only active users. An example:

   ```
   obMappingFilter="(&(objectclass=wwmOrgPerson) (uid=%loginid%) (
   | ( ! (obuseraccountcontrol=*))
   (obuseraccountcontrol=ACTIVATED)))"
   ```

   Note: This example uses the Oblix sample data (wwmOrgPerson). Change this object class to your site-specific object class. The uid=%loginid% assumes the form has a field called loginid and that this value is also included in the creds field.

Including Non-Active Users

You may want to include non-active users in your obMappingFilter so that deactivated users cannot login. To do this, you filter users with a status of obuseraccountcontrol=PENDING-ACTIVATION or PENDING DEACTIVATED.

To include only non-active users in the obMappingFilter

1. Follow the procedure “To configure a form-based authentication scheme” on page 357.
2. In the mapping filter, specify the inactive users. For example:

```
obMappingFilter="(&(objectclass=wwmOrgPerson)
(uid=%userid%)(!(|(obuseraccountcontrol=
PENDING-ACTIVATION)(obuseraccountcontrol=DEACTIVATED)
(obuseraccountcontrol=PENDING-DEACTIVATION))))"
```

Note: This example uses the Oblix sample object class wwmOrgPerson. You must change this object class to your site-specific object class. The uid=%loginid% assumes the form has a field called loginid and that this value is also included in the creds field.

Form Examples

The following sections contain examples of forms that can be used for form-based authentication:

Form Scheme Examples

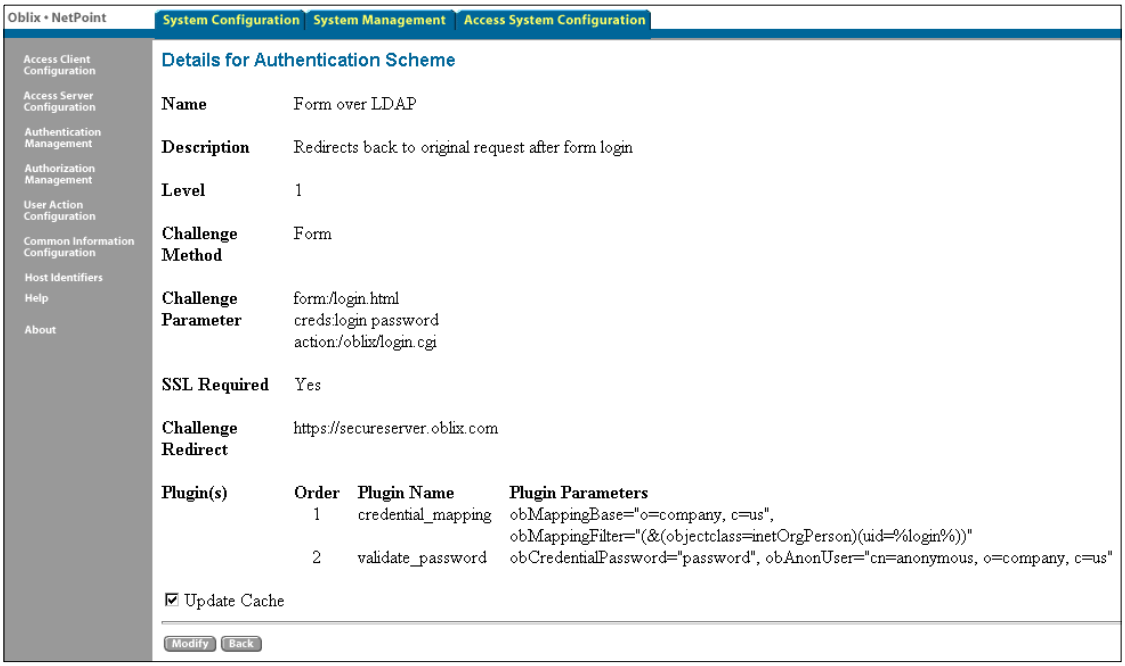
The following are examples of HTML forms and corresponding authentication schemes.

Basic Example

The following is a very simple login form:

```
<html>
<h1>My Login Form</h1>
<form name="loginform" action="/oblix/login.cgi" method="post">
Login ID: <input type="text" name="login" size="20" value="">
<p>
```

```
Submit:<input type="submit" name="submit" value="OK">
<p>
Password:<input type="password" name="password" value="">
</form>
</html>
```



Annotated Example

Figure 13 on page 364 is another sample login form. It shows the minimum requirements for a NetPoint form login authentication scheme. A production login form can be enhanced for aesthetics and branding. An example of an authentication scheme using this form is as follows:

Name—Sample NetPoint Form Login

Description—Uses SampleLoginForm.html.

Level—1

Challenge Method—Form

Challenge Parameters—

form—/loginforms/SampleLoginForm.html

creds—userid password

action—/access/oblix/apps/webgate/bin/webgate.dll

SSL Required—no

Challenge Redirect—(none)

Enabled—yes

Plug-ins—

```
credential_mapping
obMappingBase="o=Company,c=US",
obMappingFilter="(&(&(objectclass=gensiteorgperson)
(uid=%userid%))(|(!(obuseraccountcontrol=*))
(obuseraccountcontrol=ACTIVATED)))"
validate_password obCredentialPassword="password
```

For Active Directory, use “user” for the object class and “samaccountname” for the login. For example:

```
credential_mapping for Active Directory
obMappingBase="ou=Hokaido,dc=perry,dc=oblix,dc=com",
obMappingFilter="(&(&(objectclass=user)(samaccountname=%login%)
)(|(!(obuseraccountcontrol=*))(obuseraccountcontrol=ACTIVATED))
)"
```

The login form must be either unprotected or protected by an authentication scheme with a challenge method of None. This ensures that the user is not re-challenged when redirected to the login form. For the sample scheme, you can configure a policy domain that protects the form using the NetPoint None Authentication Scheme. This sets a temporary ObSSOCookie when the login form is displayed. The ObSSOCookie is rewritten after a successful login.

In the sample scheme, the userID is the uid attribute from the user’s directory profile. The credential_mapping plug-in searches the user directory from the base o=Company,c=US. The credential_mapping plug-in searches for the gensiteorgperson object that contains a uid matching the submitted userID. The additional information in the ObMappingFilter determines whether the user is activated. The validate_password plug-in performs a BIND to the directory, using the submitted password and the user profile DN retrieved when the credential_mapping plug-in searches the directory.

The action is the WebGate local URL. This URL must be protected using any NetPoint authentication scheme. For example, you might use the NetPoint Access Manager policy domain that was optionally created during setup of the Access Manager.

In the case of IIS, the WebGate action is executed as an ISAPI extension, which allows it to safely obtain the post data containing the credentials. In the case of other Web servers, WebGate intercepts the post request (because the action URL is protected) and extracts the post data for authentication. WebGate sets the ObFormLoginCookie using the action challenge parameter as its path. This ensures that the ObFormLoginCookie is returned only on the post request from the form

submission. The ObFormLoginCookie contains information about the originally requested resource. After a successful authentication, WebGate uses this information to redirect the user's browser to the originally requested resource. In the redirection, WebGate sets the ObSSOCookie with the user identity, authentication scheme level, session start and refresh time, and browser IP address.

Figure 13 Sample Login Form

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>NetPoint Sample Login Form</TITLE>
<META http-equiv=Content-Type content="text/html;
charset=windows-1252">
<META content="MSHTML 6.00.2800.1226" name=GENERATOR>
</HEAD>
<BODY>
<H2>NetPoint Sample Login Form</H2>
<FORM name=SampleLoginform action=/access/oblix/apps/webgate/bin/
webgate.dll method=post>
UserID
<INPUT name=userid>
Password
<INPUT type=password name=password>
<INPUT type=submit value=Login name=submit>
</FORM>
</BODY>
</HTML>
```

Sample Pop-Up Forms

The following JSP and ASP code samples create a pop-up login form. This prevents any issues that can arise when a login form is included as a frame within a frameset. The JSP code must be used with a Web server that has a JSP servlet engine. The ASP code must be used with IIS or another Web server with an ASP engine.

When you use one of these login pop-up examples, you need to configure an authentication scheme using one of the following challenge parameters:

form:login/login.asp (assuming the ASP form is stored under the /login folder)

or

form:login/login.jsp (for the JSP form)

Figure 14 JSP Code Sample

```
<%@ page import="java.util.*" %>
<%
int launchStatus = -1;
String URLVal = "";
String HTTPStart = "http://";
String QueryStr = request.getQueryString();
String ServerName = request.getServerName();
String PathName = request.getServletPath();
if (QueryStr != null)
{
   if (QueryStr.indexOf("launchForm") == -1)
   {
     launchStatus = -1;
   }
   else
   {
     launchStatus = 0;
   }
   URLVal = HTTPStart.concat(ServerName);
   URLVal = URLVal.concat(PathName);
   URLVal = URLVal.concat("?");
   URLVal = URLVal.concat(QueryStr);
   URLVal = URLVal.concat("&launchForm=TRUE");
}
else
{
   URLVal = HTTPStart.concat(ServerName);
   URLVal = URLVal.concat(PathName);
   URLVal = URLVal.concat("?launchForm=TRUE");
}

if ((launchStatus != 0))
{
%>
   <HTML>
   <HEAD>
   <SCRIPT Language="Javascript">
   function openLoginForm()
```

```
   {
     newUrl = "<%= URLVal %>";
     child = window.open(newUrl, "loginFormWindow",

"toolbar=no,directories=no,menubar=no,status=no,scrollbar=no,resizable=yes,widt
h=670,height=400");
     if (child.opener == null)
     {
       child.opener = window;
     }

     window.name = "loginOpener";

     if (navigator.appName == "NetScape") {
       child.focus();
     }
   }
   </SCRIPT>
   </HEAD>
   <BODY bgcolor="#ffffff" onload="openLoginForm(); return true;">
   <center>
   <p>
   <hr>
   <p>
   <Font face="verdana" size="2">
   Please enter your login credentials
   </Font>
   <p>
   <hr>
   <p>
   </center>
   </BODY>
   </HTML>

<%} else %>

   <html>
   <script language="JavaScript">
   function setAction()
```

```
    {
      document.forms[0].target=self.opener.name;
      document.forms[0].submit();
      window.close();
    }
    </script>
    <body>
    <center>
    <h1>User Login</h1>
    <br>
    <br>
    <form name="frmlogin" action="/FormProtect/login.cgi" method="post"
      target="loginOpener">
    <hr>
    <b>User ID  </b><input type="text" name="txtUserID">
    <br>
    <b>Password  </b><input type="password" name="pwdPassword">
    <br>
    <input type="button" title="Login" onclick="javascript:setAction();
      " value=Submit>
    <hr>
    </center>
    </form>
    </body>
    <html>
```

Figure 15 ASP Code Sample

```
<%
dim launchForm
launchForm = Request("launchForm")
if launchForm <> "True" then
    'This is the plain/blank HTML page
%>
    <HTML>
      <HEAD>
      <SCRIPT Language="Javascript">
        function openLoginForm()
        {
          // now open the new window with newUrl
```

```
        <% if len(request.servervariables("QUERY_STRING")) > 0 then %>
          newUrl= "<%=request.servervariables("URL") & "?" &
            request.servervariables("QUERY_STRING") & "&launchForm=True"%>";
        <% else %>
          newUrl= "<%=request.servervariables("URL") & "?launchForm=True"%>";
        <% end if %>
        child = window.open(newUrl, "loginFormWindow",

"toolbar=no,directories=no,menubar=no,status=yes,scrollbar=yes,resizable=yes,wi
dth=670,height=400");
        if (child.opener == null)
        {
          child.opener = window;
        }

        window.name = "loginOpener";

        if (navigator.appName == "NetScape")
        {
          child.focus();
        }
       }
     </SCRIPT>
     </HEAD>
     <BODY bgcolor="#ffffff" onload="openLoginForm(); return true;">
     <center>
     <p>
     <hr>
     <p>
     <Font face="verdana" size="2">
     Please enter your login credentials
     </Font>
     <p>
     <hr>
     <p>
     </center>
     </BODY>
   </HTML>
```

```
<% else %>

    <HTML>
    <SCRIPT language="JavaScript">
    function setAction()
    {
      document.forms[0].target=self.opener.name;
      document.forms[0].submit();
      window.close();
    }
    </SCRIPT>
    <BODY>
    <CENTER>
    <H1>User Login</H1>
    <BR>
    <BR>
    <form name="frmlogin" action="/FormProtect/login.cgi"
       method="post" target="loginOpener">
    <HR>
    <B>User ID  </B><input type="text" name="txtUserID">
    <BR>
    <B>Password  </B><input type="password" name="pwdPassword">
    <BR>
    <input type="button" title="Login" onclick="javascript:setAction();
       " value=Submit>
    <HR>
    </CENTER>
    </FORM>
    </BODY>
    <HTML>
<% end if %>
```

Sample Multi-Language Form

The following ASP code sample is a multi-language form that supports both Spanish and English.

Figure 16 Multi-Language Form

```
<%
Option explicit
dim strLanguage, strNewLanguage, intPointer
dim bolLoginToNetPoint
bolLoginToNetPoint = Request("LoginToNetPoint")
if bolLoginToNetPoint = true or bolLoginToNetPoint = "true" then
   bolLoginToNetPoint = true
else
   bolLoginToNetPoint = false
end if

strLanguage = Request.Cookies("PrefLang")
'Response.Write "lenguaje:" & strLanguage
if strLanguage = "" or strLanguage = "EN" then
   strLanguage = "EN"
   strNewLanguage = "SP"
   intPointer = 0
else
   strLanguage = "SP"
   strNewLanguage = "EN"
   intPointer = 1
end if

dim strUser(1),strPassword(1),strEnter(1),strPreferences(1),strCancel(1)
dim
strLanguageDescription(1),strForgot(1),strDescription(1),strMsgUandP(1),strMsgU
(1)
dim strUserType(1),strNetPointUser(1),strNetPointAdmin(1)

strUser(0) = "User:"
strUser(1) = "Usuario:"
strPassword(0)="Password:"
strPassword(1)="Contraseña:"
strEnter(0) = "Enter"
```

```
strEnter(1) = "Proceder"
strPreferences(0)="Preferences"
strPreferences(1)="Preferencias"
strCancel(0)="Cancel-Portada"
strCancel(1)="Cancelar-Portada"
strLanguageDescription(0)="Espanol"
strLanguageDescription(1)="English"
strForgot(0)="Forgot your password?"
strForgot(1)="¿Olvidó su contraseña?"
strMsgUandP(0)= "Please enter your user name and password."
strMsgUandP(1)= "Por favor teclee su usuario y contraseña."
strMsgU(0)= "Please enter your user name."
strMsgU(1)= "Por favor teclee su usuario."
strUserType(0) = "User Type:"
strUserType(1) = "Tipo de Usuario:"
strNetPointUser(0) = "NetPoint User"
strNetPointUser(1) = "Usuario NetPoint"
strNetPointAdmin(0)= "NetPoint Admin"
strNetPointAdmin(1) = "Administrador NetPoint"

strDescription(0)="Click ""Preferences"" to see and modify some of your
attributes." & _
          "<p>Da un clic en ""Español"" para cambiar esta pagina de idioma." & _
           "<p>Click ""Forgot your password?"" if you don't remember your
               password and you need to change it, " & _
           "you will be prompt to answer yor challenge phrase."

strDescription(1)="Da un clic en ""Preferencias"
             " para ver y modificar algunos de tus atributos." & _
           "<p>Click on ""English"" to change the language of this page." & _
           "<p>Da un clic en ""¿Olvidó su contraseña?"
            " si no recuerdas tu clave y deseas cambiarla, " & _
           "será necesario contestar tu frase personal."

dim identityProgram
dim userDN
dim finalURL

identityProgram="/identity/oblix/apps/userservcenter/bin/
userservcenter.cgi?program=modify&tab_id=Employees"
```

```
userDN = Request.ServerVariables("HTTP_USERDN")
finalURL = identityProgram & "&uid=" & userDN

dim obTemp
dim ObSSO
dim ObLogin
ObSSO = "ObSSOCookie=loggedout; path=/; domain=.oblix.com"

Response.Cookies("ObFormLoginCookie") = "done 1"
Response.Cookies("ObFormLoginCookie").Expires = Date() - 1

obtemp = "ObTEMP=%23comp_cookie=false%23; path=/"

%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- saved from url=(0018)http://10.26.3.90/ -->
<HTML><HEAD>
<TITLE>Login</TITLE>
<meta http-equiv="pragma" content="no-cache">
<%if bolLoginToNetPoint then%>
<meta http-equiv="Set-Cookie" content="<%=ObLogin%>"
<%end if%>

<META http-equiv=Content-Type content="text/html; charset=windows-1252">

<SCRIPT LANGUAGE=javascript>
<!--

   //Functions for keydown

   nextfield = "login";
   netscape = "";
   ver = navigator.appVersion; len = ver.length;
   for(iln = 0; iln < len; iln++) if (ver.charAt(iln) == "(") break;
   netscape = (ver.charAt(iln+1).toUpperCase() != "C");

   function keyDown(DnEvents) {
     k = (netscape) ? DnEvents.which : window.event.keyCode;
     if (k == 13) {
```

```
    if (nextfield == 'done')
    {
      setAction();
  }else{
      eval('document.loginform.' + nextfield + '.focus()');
    return;
  }
  }
 }
document.onkeydown = keyDown;
if (netscape) document.captureEvents(Event.KEYDOWN|Event.KEYUP);
//\Functions for keydown

expireDate = new Date
expireDate.setFullYear(expireDate.getFullYear()+7)
var URLs = new Array(2);
URLs[0] = "/identity/oblix/apps/userservcenter/bin/
   userservcenter.cgi?usertype=delegatedIdentityAdminBIZ";
URLs[1] = "/identity/oblix/apps/admin/bin/
   front_page_admin.cgi?usertype=systemAdmin";

function setCookie (name, value, expires) {
    document.cookie = name + "=" + escape (value)
     + "; expires=" + expireDate.toGMTString() +  "; path=/";
}
function delCookie (name) {
    var expireNow = new Date();
    document.cookie = name + "=" + "; expires=Thu,
     01-Jan-70 00:00:01 GMT" +  "; path=/";
}

function changeLanguage(){
  setCookie("cemexPrefLang","<%=strNewLanguage%>");
  document.location.reload(true);
}

// Delete the cookie
function deletecookie(){
  if (document.cookie != "") {
    thisCookie = document.cookie.split("; ")
```

```
      expireDate = new Date
       expireDate.setDate(expireDate.getDate()-1)
      for (i=0; i<thisCookie.length; i++) {
       cookieName = thisCookie[i].split("=")[1]
           document.cookie = "cookieName="+cookieName + ";expires=" +
              expireDate.toGMTString();
      }
  }
}

function killObCokies(){
  // Kill Any Cookies...

  document.cookie = "<%=obSSO%>"
     document.cookie = "<%=obLogin%>"

  //document.cookie = "ObTEMP=; path=/";
  //delCookie("ObSSOCookie");
  //delCookie("ObFormLoginCookie");
  //delCookie("ObTEMP");
}

function mySubmit() {
  if (!((loginform.login.value.length > 0) &&
    (loginform.password.value.length > 0))) {
    alert("<%=strMsgUandP(intPointer)%>");
    return;
  }
  // Kill Any Cookies...
  killObCokies();
  //document.cookie = "ObSSOCookie=loggedout; path=/;
    domain=.cemexnetlab.com"
  document.location.reload(true);


  document.cookie = "<%=obTemp%>";


  myWindowHandle = window.open
    ('about:blank','myWindowName','scrollbars=yes,width=600,height=500');
  loginform.action="/identityredirect/redirector.asp";
```

```
    //loginform.action="/identity/oblix/apps/userservcenter/bin/
       userservcenter.cgi?program=modify&usertype=endUser";
      loginform.target="myWindowName";
      loginform.submit();


    }


    function setAction() {
      if (!((loginform.login.value.length > 0) &&
       (loginform.password.value.length > 0)))
      {
       alert("<%=strMsgUandP(intPointer)%>");
       document.loginform.login.focus();
       return;
      }
      killObCokies();   // Kill Any Cookies...
      document.cookie = "<%=obTemp%>";


    <%if bolLoginToNetPoint then%>
      loginform.action = eval
("URLs["+loginform.selectName.options[loginform.selectName.selectedIndex].value
+"]");
    <%else%>
      loginform.action="/access/oblix/apps/webgate/bin/webgate.dll";
    <%end if%>
      loginform.target="";
      loginform.submit();
    }


    function lost() {
      if (!(loginform.login.value.length > 0)) {
       alert("<%=strMsgU(intPointer)%>");
       return;
      }
      // Kill Any Cookies...
      killObCokies();


      myWindowHandle = window.open
       ('about:blank','myWindowName','scrollbars=yes,width=600,height=500');
      loginform.action="/identity/oblix/apps/lost_pwd_mgmt/bin/
       lost_pwd_mgmt.cgi";
```

```
      loginform.target="myWindowName";
      loginform.submit();
    }

-->
</SCRIPT>

</HEAD>

<BODY leftMargin=0 topMargin=0 scrolling="no">

<%if bolLoginToNetPoint then%>
    <form name="loginform" action="/identity/oblix/apps/userservcenter/bin/
       userservcenter.cgi?usertype=delegatedIdentityAdminBIZ" method="post">
<%else%>
    <form name="loginform" action="/access/oblix/apps/webgate/bin/webgate.dll"
       method="post">
<%end if%>

<input type="hidden" name="ObLoginDomain" value="dc=oblix,dc=com">

<TABLE cellSpacing=0 cellPadding=0 width=763 align=center border=0>
  <TBODY>
  <TR vAlign=top>
    <TD width="39%" colSpan=2>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
        <TBODY>
        <TR>
          <TD vAlign=top width="99%" bgColor=#cc0033>
            <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
              <TBODY>
              <TR>
                <TD align=left><B><FONT face="Verdana, Arial, Helvetica,
                    sans-serif" color=#ffffff size=2>Login</FONT></B> </TD>
     </TR></TBODY>
             </TABLE>
            </TD>
        </TR>
        <TR>
```

```
<TD>
  <TABLE cellSpacing=0 cellPadding=0 width=255 border=0>
    <TBODY>
    <TR>
      <TD align=middle colSpan=3>
        <TABLE width="90%" border=0>
          <TBODY>
          <TR>
            <TD align=right width="50%"><FONTclass=classBold>
             <%=strUser(intPointer)%> </FONT> </TD>
            <TD align=left width="50%">

            <input type="text" name="login" size="16"
             onFocus="nextfield='password';" value="">

          </TD></TR>
          <TR>
            <TD align=right width="50%"><%=strPassword(intPointer)%>
              </TD>
            <TD align=left width="50%">

            <input type="password" name="password"
             onFocus="nextfield='done';" value="" size="16">
            </TD></TR>
            <%if bolLoginToNetPoint then%>
<TR>
            <TD align=right width="50%">
             <%=strUserType(intPointer)%> 
            </TD>
            <TD align=left width="50%">
 <select name="selectName">
     <option selected value="0"><%=strNetPointUser(intPointer)%>
       </option>
     <option value="1"><%=strNetPointAdmin(intPointer)%>
       </option>
 </select>
            </TD></TR>
            <%end if%>
          <TR>
            <TD colSpan=2>
```

```
                    <A href="javascript:setAction();"><%=strEnter(intPointer)%>
                      </A>  
                     <A href="javascript:mySubmit();">
                      <%=strPreferences(intPointer)%></A>  
                     <A href="javascript:loginform.reset();">
                      <%=strCancel(intPointer)%></A>  
                     </TD></TR>
                   <TR>
                     <TD colSpan=2> </TD></TR>
                   <TR>
                     <TD align=right colSpan=2>
                     <A href="javascript:changeLanguage();">
                      <%=strLanguageDescription(intPointer)%></A>
                     </TD></TR>


                   <TR>
                     <TD class=classNormal align=right colSpan=2>


                    <A href="javascript:lost()"><%=strForgot(intPointer)%></A>


                   </TD></TR></TBODY>
                </TABLE></TD></TR></TBODY>
               </TABLE></TD></TR></TBODY>
              </TABLE></TD>
    <TD width="1%"></TD>
    <TD width="60%">
      <TABLE border=0>
        <TBODY>
        <TR>
          <TD class=classBold width="100%">
            <P>
            <%=strDescription(intPointer)%>
            </TD></TR></TBODY></TABLE>

</FORM>
<DIV id=logoQA><IMG src="login_files/QA.gif">
</DIV>

<SCRIPT LANGUAGE=javascript>
<!--
```

```
document.loginform.login.focus();
-->
</SCRIPT>
</BODY>
</HTML>
```

Troubleshooting

This section describes two symptoms you may encounter when implementing form-based authentication and offers suggestions to remedy those symptoms.

Symptom—The login form repeatedly reappears after the user enters credentials.

Actions—Make sure the credentials in the creds challenge parameter for the form scheme match the input fields in the form.

Make sure the authentication plug-ins for the form scheme are correct.

If you are using IIS and the form action is not the webgate.dll, make sure the WebGate filter post processing is enabled by the Registry entry

```
HKEY_LOCAL_MACHINE\SOFTWARE\Oblix\Oblix NetPoint\version\WebGate\postdata="yes"
```

where *version* is the version number of the installed NetPoint product.

To make sure that the authentication scheme is set properly, you can attempt to access a resource protected with that authentication scheme, adding the credentials as query string parameters. This simulates a form whose method is GET without actually using the form.

For example, suppose the authentication scheme uses the following creds challenge parameter:

```
creds:login password
```

If the protected URL is http://server/protected/page.html, you could launch a browser instance and type the following:

http://server/protected/page.html?login=jsmith&password=MyPwd

If jsmith and MyPwd are valid credentials, after you press Enter the page is displayed instead of the login form if the authentication scheme is working correctly but something is wrong in the form's HTML code or in the registry (in the case of IIS servers).

To verify whether a user has a valid session, you can type the following in the browser's location:

```
javascript:altert(document.cookie)
```

The pop-up window that results should contain the current cookies associated with the browser, including the ObSSOCookie. This can also help determine if the cookie domain or invalid logout situations are affecting the login process.

Symptom—After you submit the login form, you get one or more of the following messages:

- 500 Internal Server Error
- An additional login challenge (for example, basic login dialog box)

Actions—Make sure the form's action URL is protected by a NetPoint policy domain.

Make sure the action challenge parameter of the form scheme matches the form action URL.

Note: Because of the way NetPoint updates WebGate caches, a corrected authentication scheme is not available until after that WebGate has made another request to the Access Server. Consequently, a form login problem may occur one more time after it is corrected.

Symptom—The form stops responding after successful authentication of a user.

Actions—Be sure that the redirection action does not send the user back to the same login form.

B Enabling Impersonation with NetPoint

In a Windows environment, all processes and threads execute in a security context. Impersonation is the ability of a thread to execute in a security context that is different from that of the process that owns the thread. The primary purpose of impersonation is to trigger access checks against a client's identity. For details about enabling impersonation in NetPoint, which overrides impersonation enabled with IIS, see the following discussions:

Note: The *NetPoint Integration Guide* provides a detailed example of how to integrate NetPoint with the SharePoint Portal Server as well as the extra measures you may have to take to get impersonation running in different contexts.

About Windows Impersonation

When running in a client's security context, a service becomes the client to an extent. One of the service's threads uses an access token (a protected object that represents the client's credentials) to obtain access to objects the client can access.

The client's access token is known as an impersonation token. The impersonation token identifies the client, the client's groups, and the client's privileges. The information in the token is used during access checks when the thread requests access to resources on the client's behalf. When the server is impersonating the client, any operations performed by the server are performed using the client's credentials.

Impersonation ensures that the server can do exactly what the client can do. This means that access to resources may be either restricted or expanded, depending on what the client has permission to do. Impersonation requires the participation of both the client and the server. The client must indicate its willingness to let the server use its identity, and the server must explicitly assume the client's identity programmatically. Impersonation does not allow the server to access remote resources on behalf of the client.

When impersonation concludes, the thread uses the primary token to operate using the service's own security context rather than the client's. The primary token describes the security context of the user account associated with the process (the person who started the application).

Services run under their own accounts and act as users in their own right. For example, system services that are installed with the operating system run under the Local System account. You can configure other services to run under the Local System account, or separate accounts on the local system or in Active Directory.

The IIS Web server provides impersonation capabilities. However, NetPoint overrides IIS authentication, authorization, and impersonation functions. For more information, see:

- "About Impersonation and NetPoint" on page 383
- "Windows Impersonation Background" on page 401

SSO for Authenticated NetPoint Users into Exchange/Outlook Web Access (OWA)— This is also supported using the Windows Impersonation feature. OWA provides Web access to Exchange mail services and may be configured on either of the following:

- An IIS Web server that does not reside on the same server as the Exchange server, which is also known as a "front-end" server
- An IIS Web server running on the Exchange server, which is also known as the "back-end" server

In a "front-end" server configuration, the front-end OWA server authenticates the user, determines the back-end Exchange server that hosts the user's mailbox, then proxies the request to the appropriate back-end Exchange server. No additional credential information is passed. No delegation is performed. Setting up Impersonation on the back-end Exchange server ensures that the Exchange server does not need to request credentials before granting access.

For more information, see "Setting Up Impersonation for OWA" on page 395.

About Impersonation and NetPoint

You can enable NetPoint support for Windows impersonation to provide additional access control to protected applications. To ensure success, you need to bind a trusted user to a NetPoint WebGate and the application must be protected by a NetPoint policy domain that includes an impersonation action in the authorization rule. Authentication will occur as usual. However, during the authorization process, the protected application will create an impersonation token.

Table 22 identifies NetPoint support for Windows impersonation.

Table 22 NetPoint Support for Windows Impersonation

NetPoint v6.5 and Later Supports	Previous NetPoint Versions Supported
Microsoft Kerberos Service-for-User-to-Self (S4U2Self) extension	Username and password required. LOGON_USER, LOGON_PASSWORD (in authorization rule, action)
"Impersonate" HeaderVar action type is as an Authorization Rule Action in NetPoint	Username (LOGON_USER) used in proper header variables.
No password needed	Password (LOGON_PASSWORD) stored in a directory in clear text or in a separate database, not set as a header variable.
REMOTE_USER may be set to any value (in Authorization Rule, Action (type HTTP).	No change

For more information, see "The Kerberos Protocol" on page 403 and "The S4U2Self Extension" on page 404. Also, see the following:

- "Enabling Impersonation With a Header Variable" on page 384 provides prerequisites and details about implementing impersonation using header variables.
- "Enabling Impersonation with a User Name and Password" on page 394 explains how to implement impersonation in NetPoint using features available *before* NetPoint 6.5.

Enabling Impersonation With a Header Variable

Enabling impersonation with a header variable in NetPoint involves the following procedures.

Task overview: Enabling impersonation with a header variable includes

1. Reviewing all “Requirements” on page 384
2. “Creating an Impersonator as a Trusted User” on page 385
3. “Binding the Trusted User to Your WebGate” on page 387
4. “Adding an Impersonation Action to a Policy Domain” on page 388
5. “Adding an Impersonation dll to IIS” on page 390
6. “Testing Impersonation” on page 391

Note: The example in this chapter illustrates setting up the impersonation feature for the NetPoint to Microsoft SharePoint Portal Server integration. The principles are the same regardless of your application.

See also “Setting Up Impersonation for OWA” on page 395.

Requirements

You need to prepare the environment and confirm that it is operating properly before implementing impersonation with NetPoint.

Table 23 identifies the platform requirements for NetPoint v6.5 and later when you choose to enable impersonation using a header variable.

Table 23 NetPoint v6.5 and Later Requirements for Impersonation with a Header Variable

WebGate (and Impersonation dll)	Microsoft IIS 6.x and Windows Server 2003 Note: Other NetPoint components have no specific requirements.
Impersonation dll	*WebGate_install_dir*\access\oblix\apps\webgate\bin • Must be installed as an IIS wildcard extension. • May be installed at any level of the Web site tree. For details, see “Wildcard Extension” on page 403.
Kerberos Key Distribution Center (KDC) and Active Directory	Windows Server 2003

Table 23 NetPoint v6.5 and Later Requirements for Impersonation with a Header Variable

Client and Server machines	• Both must be in the same Windows Server 2003 domain with a trust relationship. • A bidirectional trust path is required because the service, acting on the client's behalf, must request tickets from the client's domain.
Security context	Must have "Act as operating system" privileges **Note**: IWAM_*Machine* is not recommended because it is the account used by the Microsoft Transaction Server (MTS) and various IIS entities to provide programmatic and transactional functions.
Mutual authentication is required.	Mutual authentication is supported remotely

Creating an Impersonator as a Trusted User

Whether you enable impersonation using a HeaderVar or user profile attribute, the return value must be a trusted user in the Active Directory. This special user should not be used for anything other than impersonation.

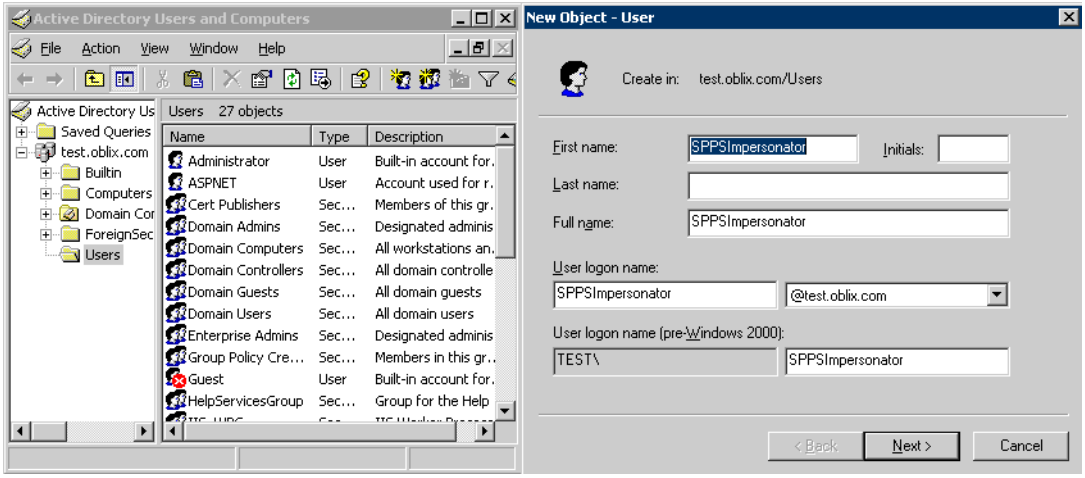
If your environment includes Exchange Outlook Web Access, see also "Setting Up Impersonation for OWA" on page 395.

To create a trusted user account

1. On the Windows 2003 machine hosting your SPPS installation, select Start > Programs > Manage Your Server > Domain Controller (Active Directory) > Manage Users and Computers in Active Directory.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New > User.
3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as *SPPSImpersonator*.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you are asked to choose a password and then retype it to confirm.

Note: Oblix recommends that you chose a very complex password, because your trusted user is being given very powerful permissions. Also, be sure to check the box marked Password Never Expires. Since the impersonation extension should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

Figure 17 Setting up a Trusted User Account for Windows Impersonation



Assigning Rights to the Trusted User

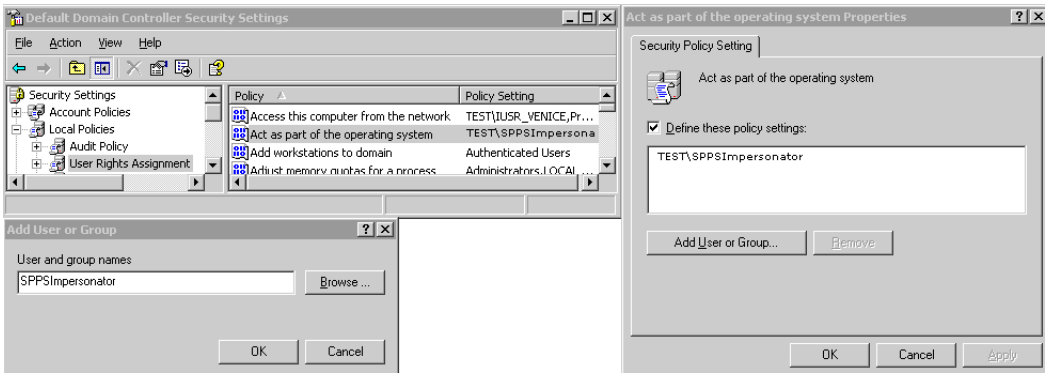
You need to give the trusted user the right to act as part of the operating system.

If your environment includes Exchange Outlook Web Access, see also “Setting Up Impersonation for OWA” on page 395.

To give appropriate rights to the trusted user

1. Select: Control Panel > Administrative Tools > Domain Controller Security Policy.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click User Rights Assignment on the tree in the left pane.
4. Double-click “Act as part of the operating system” in the right pane.
5. Click Add User or Group.
6. In the Add User or Group panel, type the User logon name of the trusted user (SPPSImpersonator in our example) in the User and group names text entry box, then click OK to register the change.

Figure 18 Configuring Rights for the Trusted User in Windows Impersonation



Binding the Trusted User to Your WebGate

You need to bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described below.

If your environment includes Exchange Outlook Web Access, see also “Setting Up Impersonation for OWA” on page 395.

To bind your trusted user to your WebGate

1. Point your browser to your Access System Console.

   For example:

   http://*hostname*.*domain*.com:*port*/access/oblix

   where *hostname* is the DNS name of the machine hosting your Access Manager, *domain* is the name of the server domain to which the machine belongs, and *port* is the number of the port to which Access Manager listens.

2. Navigate to Access System Console > Access System Configuration > AccessGate Configuration.

3. Select the name of the WebGate you want to modify.

   The Details for NetPoint AccessGate page appears with a summary of the configuration information for this WebGate. At the bottom of this Web page are fields for Impersonation Username and Impersonation Password.

4. Click the Modify button at the bottom of the page.

5. In the Modify NetPoint AccessGate page, scroll to the bottom and enter the username and password for the trusted user account you created through the task on page 385.

   For example:

Impersonation username:	SPPSImpersonator
Impersonation password:	**********
Re-type impersonation password:	**********

6. Click the Save button to commit the changes and return to the Details page.

   A bind has been created for the WebGate and the trusted user. The WebGate is now ready to provide impersonation on demand. The demand is created by an Authorization Success Action in a policy domain created for impersonation.

Adding an Impersonation Action to a Policy Domain

You must create or configure a NetPoint policy domain to protect your SharePoint resources. You do this by adding an Authorization Success Action with a return type of headerVar, the name parameter set to the name of the trusted user (SPPSImpersonator in our example), and the return attribute parameter set to samaccountname for a single-domain Active Directory installation or userPrincipalName for a multi-domain Active Directory forest.

You must also choose an easy-to-remember name for the domain, such as *ImpersonationPolicyDomain*.

For details on creating a NetPoint policy domain, see “Protecting Resources with Policy Domains” on page 95.

If your environment includes Exchange Outlook Web Access, see also “Setting Up Impersonation for OWA” on page 395.

To add an impersonation action to your policy domain

1. Point your browser to the Access System Console. For example:

   http://*hostname*.*domain*.com:port/access/oblix

   where *hostname* is the DNS name of the machine hosting your Access Manager, *domain* is the name of the server domain to which the machine belongs, and *port* is the number of the port to which Access Manager listens.

2. Navigate to the Authorization Definitions page of the policy domain you want to change:

   Access Manager > My Policy Domains > *PolicyName* > Authorization Rules

   where *PolicyName* refers to the policy domain you created specifically for impersonation (*ImpersonationPolicyDomain* in our example).

   Note: Currently defined authorization rules are listed. If none are listed, click the Add button and complete the form to create one.

3. Click the link to the rule to which you want to add the impersonation action. The description will expand.
4. Click the Actions link, which appears directly under the Authorization Rules tab.

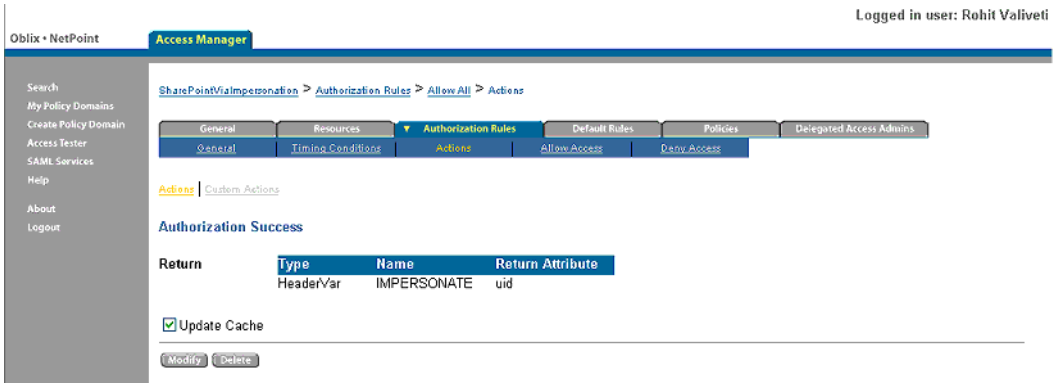
   The Authorization Success page appears. If no actions are identified, you must add them. If actions are provided, you can modify them.

   You need to add a header variable named impersonate to Authorization Success Action in the policy domain for impersonation.
5. On the Authorization Success page appears, click Add or Modify.
6. Complete the form using headerVar as the Return Type, the User log on name of the trusted user you have bound to the WebGate, and the appropriate return value for your environment.

   For example:

   Type: HeaderVar
   Name: IMPERSONATE
   Return value: *uid* or *samAccountName* (Active Directory username, the Windows domain user for the desired folder)

   Your completed form may look something like the one below.



7. Save the rule.

This rule is used for the second WebGate request (for authorization).

Adding an Impersonation dll to IIS

You are ready to configure IIS by adding the IISImpersonationExtension.dll to your IIS configuration.

To add the impersonation dll to your IIS configuration

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click Web Service Extensions on the tree in the left pane.
4. Double-click Oblix WebGate in the right panel to open the Properties panel.
5. Click the Required Files tab.
6. Click Add.
7. In the Path to file text box, type the full path to IISImpersonationExtension.dll.

   By default, the path is:

   WebGate_install_dir\access\oblix\apps\webgate\bin\IISImpersonation\Extension.dll

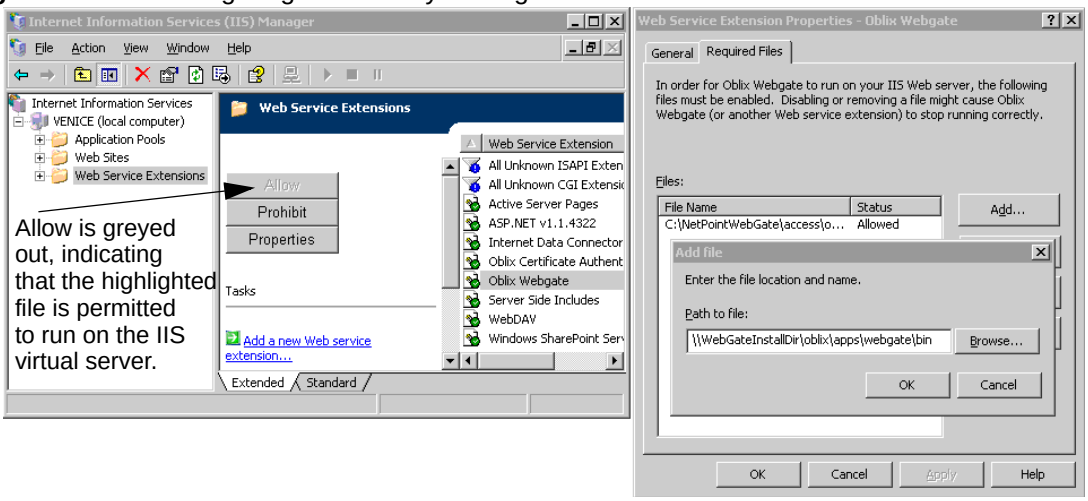
   where *WebGate_install_dir* is the root directory of your WebGate installation.

   Note: If any spaces exist in the path (for example, C:\Program Files\NetPoint\...) surround the entire string with double quotes (“ ”).

8. Click OK.
9. Click the General tab on the Web Services Extension Properties panel, then verify that the box beside “Do not check the file location” is *not* checked.
10. Verify that the Allow button to the left of the Oblix WebGate icon is greyed out, as shown below, which indicates that the dll is allowed to run as a Web service extension.

    Note: If Allow is not greyed out, click it so that it becomes greyed out.

Figure 19 Configuring IIS Security Settings



Testing Impersonation

You can test Impersonation in the following two ways:

- “Testing Impersonation Using the Event Viewer” on page 392
- “Testing Impersonation using a Web Page” on page 393

Creating an IIS Virtual Site Not Protected by SPPS

To test the impersonation feature outside the SPPS context or to test SSO, you will need a target Web page on an IIS virtual Web site that is not protected by SPPS. You create such a virtual Web site by completing the following task.

To create an IIS virtual site not protected by SPPS

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Right-click Web Sites on the tree in the left pane, then navigate to New > Web Site on the menu.
4. Respond to the prompts by the Web site creation wizard.
5. After you create the virtual site, you must protect it with a NetPoint Policy domain, as described elsewhere in this guide.

Testing Impersonation Using the Event Viewer

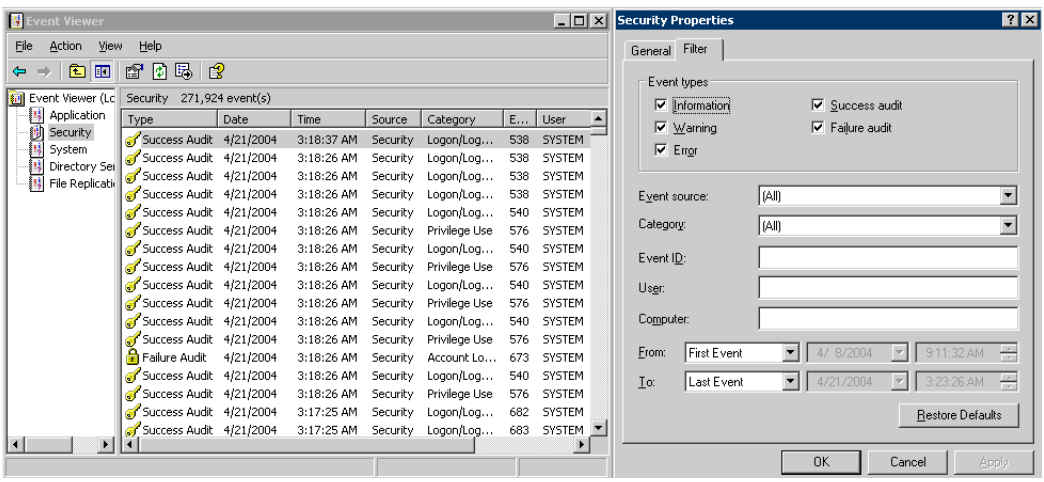
When you complete impersonation testing using the Windows 2003 Event Viewer, you must configure the event viewer before conducting the actual test.

To test impersonation through the Event Viewer

1. Select Start Menu > Event Viewer.
2. In the left pane, right-click Security, then click Properties.
3. Click the Filter tab on the Security property sheet.
4. Verify that all Event Types are checked, and the Event Source and Category lists are set to All, then click OK to dismiss the property sheet.

   Your Event Viewer is now configured to display information about the headerVar associated with a resource request.

Figure 20 Verifying Event Viewer Settings



5. Create a new IIS virtual server (virtual site).
6. Place a target Web page anywhere in the tree on the virtual site.
7. Point your browser at the Web page.

   If impersonation is working correctly, the Event Viewer will report the success of the access attempt.

Testing Impersonation using a Web Page

You can also test impersonation using a dynamic test page, such as an .asp page or a perl script, that can return and display information about the request.

To test impersonation through a Web page that displays server variables

1. Create an .asp page or perl script that will display the parameters AUTH_USER and IMPERSONATE. It can resemble the sample page presented in the following listing:

Listing 1 Sample .ASP Page Code

```
<TABLE border=1>
<TR>
   <TD>Variable</TD>
   <TD>  </TD>
   <TD>Value</TD></TR>
<%for each servervar in request.servervariables%>
<TR>
   <TD><%=servervar%></TD>
   <TD>  </TD>
   <TD><%=request.servervariables(servervar)%> </TD>
</TR>
```

2. Create an IIS virtual site, or use the one you created for the previous task.
3. Place a .asp page or perl script (such as the sample in the preceding listing) anywhere in the tree of the new virtual site.
4. Point your browser at the page. The page should display, with both AUTH_USER and IMPERSONATE set to the name of the user making the request.

Setting Up Impersonation with Integrations

The *NetPoint Integration Guide* provides a detailed example of how to integrate NetPoint with the SharePoint Portal Server (SPPS) and the extra measures you may have to take to get impersonation running in different contexts:

- Configuring IIS Security
- Configuring the Wildcard Extension
- Editing web.config (this is not needed with the integration between NetPoint and the Microsoft Content Management Server)

Enabling Impersonation with a User Name and Password

The method to enable impersonation *before* NetPoint v6.5 remains valid and may also be used with NetPoint v6.5 and later, as described below.

NetPoint provides an API that tells IIS which user to impersonate. To use this API, you must provide the user name and password to IIS. The user name is used in the proper header variables. This causes IIS to change the owner of the thread for downstream applications such as Outlook Web Access.

To have IIS log in as the user, you set the following two success actions in the authorization policy:

- **LOGON_USER**—The NT username of the user you want to impersonate
- **LOGON_PASSWORD**—The NT password of the user.

The LOGON_PASSWORD is not set as a header variable. This prevents downstream applications from learning the password. This variable is only used to impersonate the user. The following are methods for providing the Windows NT or Active Directory (AD) password:

- Store the NT or AD password in clear text in the directory, then configure the NetPoint security policy to set the proper header variable with the password value.
- Store the password in a separate database.

  This requires an authorization plug-in to be written to access the password and set the appropriate header variable. The authorization plug-in supplies the action with the password. The store would have to be kept synchronized using the COREid System's PPP mechanisms.
- Create a static header variable that impersonates the user for a particular role (for instance, manager) that provides the proper security settings. This provides a more granular option if you do not require the actual individual to be impersonated.

NetPoint supports additional IIS header variables for integration with Microsoft environments and Windows Impersonation, as shown in Table 24.

Table 24 NetPoint Supports Additional IIS Header Variables

REMOTE_USER	AUTH_USER	AUTH_PASSWORD	AUTH_TYPE

These are special case headers that show downstream applications that the user is logged in. If you set the REMOTE_USER header by creating a REMOTE_USER http header action, NetPoint will set the AUTH_USER and AUTH_PASSWORD

headers. You set REMOTE_USER in the same place as LOGON_USER and LOGON_Password, as a success action in the authorization policy. Setting this action accomplishes the following for each of the variables:

- The REMOTE_USER will contain the static or attribute value
- The AUTH_USER will have the same value
- The AUTH_PASSWORD header will always contain HiddenByOblixNetpoint so the password remains hidden
- The AUTH_TYPE header will contain Basic

For more information:

- See “Windows Impersonation Background” on page 401 for an introduction to access tokens, security IDs, access control lists, wildcard extensions, and Kerberos.
- See the Microsoft documentation for details about single sign-on integration through Windows Impersonation.

Setting Up Impersonation for OWA

In a distributed Exchange/OWA SSO environment, each server needs NetPoint to impersonate the current user. When you enable Impersonation, you need to include additional HTTP Headers in “Authorization Success” for your impersonation policy domain:

The following solution has been tested in both standalone and distributed OWA environments.

Task overview: Setting up impersonation for OWA

1. Install NetPoint, including a NetPoint WebGate on the OWA front-end server and on all Exchange back-end servers, as described in the *NetPoint 7.0 Installation Guide*.
2. Disable IP Checking for the WebGates on the back-end server using the WebGateStatic.lst file (because the request comes from the front-end server, *not* from the user’s browser).
3. Create a trusted user account for only impersonation in the Active Directory, as described in “Creating a Trusted User Account for OWA” on page 396.
4. Give the trusted user the special right to act as part of the operating system., as described in “Assigning Rights to the OWA Trusted User” on page 396.
5. Bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described in “Binding the Trusted OWA User to Your WebGate” on page 397.

6. Add a header variable named impersonate to Authorization Success Action in the policy domain for impersonation, as described in “Adding an Impersonation Action to a Policy Domain” on page 398.
7. Configure IIS by adding IISImpersonationExtension.dll to your IIS configuration, as described in “Adding an Impersonation dll to IIS” on page 399.
8. Test Impersonation, as described in “Testing Impersonation for OWA” on page 400.

Creating a Trusted User Account for OWA

This special user should not be used for anything other than impersonation.

Oblix recommends that you chose a very complex password, because your trusted user is being given very powerful permissions. Also, be sure to check the box marked Password Never Expires. Since the impersonation extension should be the only entity that ever sees the trusted user account, it would be very difficult for an outside agency to discover that the password has expired.

To create a trusted user account for OWA

1. On the Windows 2003 machine, select Start > Programs > Manage Your Server > Domain Controller (Active Directory) > Manage Users and Computers in Active Directory.
2. In the Active Directory Users and Computers window, right-click Users on the tree in the left pane, then select New > User.
3. In the First name field of the pane entitled New Object - User, enter an easy-to-remember name such as OWAImpersonator.
4. Copy this same string to the User logon name field, then click Next.
5. In succeeding panels, you will be asked to choose a password and then retype it to confirm.
6. Proceed to “Assigning Rights to the OWA Trusted User” on page 396.

Assigning Rights to the OWA Trusted User

You need to give the trusted user the right to act as part of the operating system.

To give appropriate rights to the trusted user

1. Select Control Panel > Administrative Tools > Domain Controller Security Policy.
2. On the tree in the left pane, click the plus icon (+) next to Local Policies.
3. Click User Rights Assignment on the tree in the left pane.

4. Double-click “Act as part of the operating system” in the right pane.
5. Click Add User or Group.
6. In the Add User or Group panel, type the User logon name of the trusted user (OWAImpersonator in our example) in the User and group names text entry box, then click OK to register the change.
7. Proceed to “Binding the Trusted User to Your WebGate” on page 387.

Binding the Trusted OWA User to Your WebGate

You need to bind the trusted user to the WebGate by supplying the authentication credentials for the trusted user, as described below.

To bind your trusted OWA user to your WebGate

1. Point your browser to your Access System Console. For example:

   http://hostname.*domain*.com:*port*/access/oblix

   where *hostname* is the DNS name of the machine hosting your Access Manager; *domain* is the name of the server domain to which the machine belongs; and *port* is the number of the port to which Access Manager listens.
2. Navigate to Access System Console > Access System Configuration > AccessGate Configuration.
3. Select the name of the Webgate you want to modify.

   The Details for NetPoint AccessGate page appears with a summary of the configuration information for this WebGate. At the bottom of this Web page are fields for Impersonation Username and Impersonation Password.
4. Click the Modify button at the bottom of the Details for NetPoint AccessGate page.

   The Modify NetPoint AccessGate page appears.
5. Scroll to the bottom and enter the username and password for the trusted user account you created (OWAImpersonator).
6. Click the Save button to commit the changes and return to the Details page.

   A bind has been created for the WebGate and the trusted user. The WebGate is now ready to provide impersonation on demand. The demand is created by an Authorization Success Action in a policy domain created for impersonation.
7. Proceed to “Adding an Impersonation Action to a Policy Domain” on page 388

Adding an Impersonation Action to a Policy Domain

You must create or configure a NetPoint policy domain to protect your OWA resources. This policy must set several HTTP Header variables.

Note: You should choose an easy-to-remember name for the domain, such as ImpersonationPolicyDomain.

To add an impersonation action to your policy domain

1. Navigate to the Access System Console and log in. For example:

   http://*hostname*.*domain*.com:*port*/access/oblix

   where *hostname* is the DNS name of the machine hosting your WebPass and Access Manager; *domain* is the name of the server domain to which the machine belongs; and *port* is the number of the port to which Access Manager listens.

2. Navigate to the Authorization Definitions page of the policy domain you want to change:

   Access Manager > My Policy Domains > *PolicyName* > Authorization Rules

   where *PolicyName* refers to the policy domain you created specifically for impersonation (ImpersonationPolicyDomain in this example).

3. Currently defined authorization rules are listed. If none are listed, click the Add button and complete the form to create one.

4. Click the link to the rule to which you want to add the impersonation action to expand the description.

5. Click the Actions tab, directly under the Authorization Rules tab.

   The Authorization Success page appears, with a separate section for Authorization Success and Authorization Failure. If no actions are identified, you must add them. If actions are provided, you can modify them.

   You need to add header variables named “impersonate”, “auth_type”, “remote_user”, and “npusername” to the Authorization Success Action in the policy domain for impersonation.

6. On the Authorization Success page, click the Add or Modify button.

7. In the Authorization Success area, fill in the Return details.

   Type: HeaderVar
   Name: IMPERSONATE
   Return value: uid (or samaccountname)

   Type: HeaderVar
   Name: AUTH_TYPE
   Return value: NTLM

   Type: HeaderVar
   Name: REMOTE_USER
   Return value: uid (or samaccountname)

   Type: HeaderVar
   Name: NPUSERNAME
   Return value: uid (or samaccountname)

8. Save the rule, which is used for the second WebGate request for authorization.
9. Proceed with "Adding an Impersonation dll to IIS" on page 390.

Adding an Impersonation dll to IIS

You are ready to configure IIS by adding the IISImpersonationExtension.dll to your IIS configuration.

To add the impersonation dll to your IIS configuration

1. Select Start > Administrative Tools > Internet Information Services (IIS) Manager.
2. Click the plus icon (+) to the left of the local computer icon on the tree in the left pane.
3. Click Web Service Extensions on the tree in the left pane.
4. Double-click Oblix WebGate in the right panel to open the Properties panel.
5. Click the Required Files tab.
6. Click Add.
7. In the Path to file text box, type the full path to IISImpersonationExtension.dll.

   For example, by default, the path is:

   WebGate_install_dir\access\oblix\apps\webgate\bin\IISImpersonation\Extension.dll

   where WebGate_install_dir is the directory of your WebGate installation.

Note: If any spaces exist in the path (for example, C:\Program Files\NetPoint\...) surround the entire string with double quotes (" ").

---

8. Click OK.
9. Click the General tab on the Web Services Extension Properties panel, then verify that the box "Do not check the file location" is not checked.
10. Verify that the Allow button to the left of the Oblix WebGate icon is greyed out, as shown below, which indicates that the dll is allowed to run as a Web service extension.

    Note: If Allow is not greyed out, click it so that it becomes greyed out.

11. Proceed to “Testing Impersonation for OWA” on page 400.

Testing Impersonation for OWA

The following options are provided to test the Impersonation configuration for OWA.

- “Testing Impersonation Using the Event Viewer” on page 400
- “Testing Impersonation using a Web Page” on page 401

Testing Impersonation Using the Event Viewer

To test impersonation through the Event Viewer

1. Select Start Menu > Event Viewer.
2. In the left pane, right-click Security, then click Properties.
3. Click the Filter tab on the Security property sheet.
4. Verify that all Event Types are checked, and the Event Source and Category lists are set to All, then click OK to dismiss the property sheet.
5. Your Event Viewer is now configured to display information about the headerVar associated with a resource request.
6. Create a new IIS virtual server (virtual site).
7. Place a target Web page anywhere in the tree on the virtual site.
8. Point your browser at the Web page.

   If impersonation is working correctly, the Event Viewer will report the success of the access attempt.

Testing Impersonation using a Web Page

You can also test impersonation using a dynamic test page, such as an .asp that can return and display information about the request.

To test impersonation through a Web page

1. Create an .asp page or perl script that will display the parameters AUTH_USER and IMPERSONATE, which can resemble the sample page presented in the following listing:

```
<TABLE border=1>
<TR>
<TD>Variable</TD>
<TD>  </TD>
<TD>Value</TD></TR>
<%for each servervar in request.servervariables%>
<TR>
<TD><%=servervar%></TD>
<TD>  </TD>
<TD><%=request.servervariables(servervar)%> </TD>
</TR>
```

2. Create an IIS virtual site, or use the one you created for the previous task.
3. Place a .asp page or perl script (such as the sample in the preceding listing) anywhere in the tree of the new virtual site.
4. Point your browser at the page, which should appear, with both AUTH_USER and IMPERSONATE set to the name of the user making the request.

Windows Impersonation Background

The information here provides a simple overview of several Windows impersonation concepts. Topics include:

For more information, see your Microsoft documentation.

Access Tokens

The access token describes the security context of a process or thread and includes the identity and privileges of the user account associated with the process or thread. The access token is created when authentication is successful. For example:

- The logon process returns a security ID (SID) for the user and a list of SIDs for the user's security groups.
- The Local Security Authority (LSA) creates an access token that includes:
  a) The SIDs returned by the logon process
  b) A list of privileges assigned to the user and to the user's security groups by local security policy

A copy of the access token is attached to every process and thread that is executed on the user's behalf. When a thread interacts with a securable object, or tries to perform a system task that requires privileges, the operating system checks the access token associated with the thread to determine its level of authorization.

Security IDs

A security ID (SID) is a unique value of variable length used to identify a security principal or security group. SIDs are equal to NetPoint SSO tokens and represent a unique user within the Windows operating system.

The SID that identifies a particular account or group is generated by the system at the time the account or group is created. As mentioned above, the SID for a local account or group is generated by the Local Security Authority (LSA) and stored with other account information in a secure area of the registry. The SID for a domain account or group is generated by the domain security authority and stored as an attribute of the User or Group object in Active Directory.

SIDs are unique within the scope of the account or group they identify. The SID for every local account and group is unique on the computer on which it was created. No two accounts or groups on the same machine can have the same SID. The SID for every domain account and group is unique within an enterprise. The SID for an account or group created in a domain never matches the SID for any other account or group created in the same domain.

One or more SIDs are included:

- In access tokens, where one SID identifies the user represented by the token and additional SIDs identify the security groups to which the user belongs.
- In security descriptors, where one SID in an object's security descriptor identifies the object's owner and another SID identifies the owner's primary group.

- In access control entries (ACEs), the SID identifies the user or group for whom access is allowed, denied, or audited.

Access Control Lists and Entries

An access control list (ACL) contains an ordered list of access control entries (ACEs) that define the policies used to control access to resources, such as directories and applications protected by NetPoint.

All ACLs are based on your logon identity. An object's security descriptor can contain two ACLs:

- A discretionary access control list (DACL) that identifies the users and groups who are allowed or denied access
- A system access control list (SACL) that controls how access is audited

Each ACE includes:

- The type of the ACE (generic vs. object specific)
- Child-object inheritance attributes
- Access rights
- A SID that identifies a user or group

Wildcard Extension

The Web server normally runs in a security context called “IWAM_xxx” This security context does not have rights to impersonate another user. NetPoint designates a special user that does have the right to impersonate another user by configuring it using the impersonation username/password on the AccessGate configuration page. That designated user must have “act as operating system” rights, as explained elsewhere.

The wildcard extension for the impersonation DLL behaves like a filter, which means that the wildcard extension is enabled for each request to the Web server. The DLL executes after WebGate, after all filters, and before any downstream applications.

The Kerberos Protocol

The Kerberos protocol defines how clients interact with a network authentication service. The client obtains a ticket from the Kerberos Key Distribution Center (KDC). The Kerberos ticket represents the client's network credentials. The ticket is presented to a server when the connection is established.

The Kerberos protocol handles all domain lookups in all trusted domains. As the client’s identity, this protocol uses:

- The Active Directory domain name
- User name
- Password

The initial ticket that is obtained from the KDC when the user logs on is based on an encrypted hash of the user's password. This initial ticket is cached.

When the user tries to connect to a server, the Kerberos protocol checks the ticket cache for a valid ticket for that server. If one is not available, the initial ticket for the user is sent to the KDC along with a request for a ticket for the specified server. That session ticket is added to the cache and can be used to connect to the same server until the ticket expires.

The S4U2Self Extension

Windows Server 2003 domain controllers accept a new type of Kerberos request, the Service-for-User-to-Self (S4U2Self) extension. This extension enables the service to request a ticket from the client to itself, presenting its own credentials instead of the client's.

Index

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W