Novell
Access Manager

3.0 SP3

www.novell.com

ADMINISTRATION GUIDE

Interim Release 2

June 4, 2008

Legal Notices

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

Third-Party Materials

Contents

41 Troubleshooting the SSL VPN 661

42 Using the Log Files for Troubleshooting 669

43 Troubleshooting XML Validation Errors 683

44 Troubleshooting Certificate Issues 691

About This Guide

This guide describes the features of Novell® Access Manager, including:

- Part I, "System Management," on page 23
- Part II, "Novell Identity Server Configuration," on page 53
- Part III, "Access Gateway Configuration," on page 197
- Part IV, "SSL VPN Gateway Configuration," on page 311
- Part V, "Security and Certificate Management," on page 353
- Part VI, "Policy Management," on page 383
- Part VII, "Monitoring Access Manager Components," on page 503
- Part VIII, "Troubleshooting," on page 575
- Part IX, "Appendixes," on page 695

This administration guide is intended to help you understand and configure all of the features provided by Access Manager, and includes advanced topics.

It is recommended that you first become familiar with the information in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*, which helps you understand how to perform a basic Identity Server configuration, set up a resource protected by an Access Gateway, and configure SSL.

The basic setup and the administration guides are designed to work together, and important information and setup steps are not necessarily repeated in both places.

Audience

This guide is intended for Access Manager administrators. It is assumed that you have knowledge of evolving Internet protocols, such as:

- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Security Assertion Markup Language (SAML)
- Public Key Infrastructure (PKI) digital signature concepts and Internet security
- Secure Socket Layer/Transport Layer Security (SSL/TSL)
- Hypertext Transfer Protocol (HTTP and HTTPS)
- Uniform Resource Identifiers (URIs)
- Domain Name System (DNS)
- Web Services Description Language (WSDL)

Feedback

We want to hear your comments and suggestions about this guide and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to Documentation Feedback (http://www.novell.com/documentation/feedback.html) at www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the *Access Manager Administration Guide*, visit the Novell Access Manager Documentation Web site (http://www.novell.com/documentation/novellaccessmanager).

Additional Documentation

Before proceeding, you should be familiar with the *Novell Access Manager 3.0 SP3 IR2 Installation Guide* and the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*, which provides information about setting up the Access Manager system.

If you are unfamiliar with SAML 1.1, see "SAML Overview" (http://www.novell.com/documentation/saml/saml/data/ag8qdk7.html) on the Documentation Web site (http://www.novell.com/documentation/a-z.html).

For conceptual information about Liberty, and to learn about what is new for SAML 2.0, see Appendix A, "About Liberty," on page 697.

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux* or UNIX*, should use forward slashes as required by your software.

System Management

I

This section discusses system and server management topics that apply to the Novell® Access Manager Administration Console.

Security Considerations

1

This section describes some security checks that you can use to help verify the security of your Novell® Access Manager configuration.

For firewall information, see "Setting Up Firewalls" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

1.1 Certificates

Your security deployment plan should contain policies for the following:

- **Key size for certificates:** The Access Manager product ships with a CA that can create certificates with a key size of 2048, which is the maximum size supported by older software. For information about increasing the key size to 4096, see Section 24.8, "Enabling 4096k Keys," on page 370.
- **Certificate renewal dates:** We recommend that certificates should be renewed every two years. Your security needs might allow for a longer or shorter period.
- **Trusted Certificate Authorities:** The Access Manager ships with a CA, and during installation of the various components, it creates and distributes certificates. If this CA is not on your list of trusted CAs, you need to add certificates created by your trusted CAs. See Chapter 25, "Assigning Certificates to Access Manager Devices," on page 375.

1.2 Access Manager Administration Console

The admin user you create when you install the Administration Console has all rights to the Access Manager components. We recommend that you protect this account by configuring the following features:

- **Password Restrictions:** When the admin user is created, no password restrictions are set. To ensure that the password meets your minimum security requirements, you should configure the standard eDirectory™ password restrictions for this account. Go to the Administration Console and click *Users*. Browse to the admin user (found in the novell container), then click *Restrictions*. For configuration help, use the *Help* button.

- **Intruder Detection:** The admin user is created in the novell policy container. To modify the intruder detection policy for this container, go to the Administration Console and click *Directory Administration > Modify Object*. Select *novell,* then click *OK*. Click *Intruder Detection*. For configuration help, use the *Help* button.

You also need to protect the Administration Console from Internet attacks. It should be installed behind your firewall.

If you install secondary consoles for redundancy, these secondary consoles should be on the same network. For a secure system, they should not be required to cross routers to communicate with each other.

Also, if you are installing the Administration Console on a separate machine, ensure that the DNS names resolve between the Identity Server and the Administration Console. This ensures that SSL security functions correctly between the Identity Server and the configuration store in the Administration Console.

1.3 Configuration Store

The configuration store is an embedded, modified version of eDirectory. It can only be backed up and restored with command line options. The backup file is not encrypted, so it should not be used to back up user accounts with their passwords. Because of this limitation, it should not be used for a user store.

You should back up the configuration store on a regular schedule, and the ZIP file created should be stored in a secure place. See Section 2, "Backing Up and Restoring Components," on page 31.

In addition to backing up the configuration store, you should also install at least two Administration Consoles (a primary console and a secondary replica). This ensures that if the primary console goes down, the secondary console can keep the communication channels open between the various components. You can install up to three Administration Consoles.

1.4 Auditing and Event Notification

For a secure system, you need to set up either auditing or syslogging to notify the system administrator when certain events occur. The most important audit events to monitor are the following:

- Configuration changes
- System shutdowns and startups
- Server imports and deletes
- Intruder lockout detection (available only for eDirectory user stores)
- User account provisioning

Audit events are device-specific. To select auditing events, use the following:

- **Administration Console:** In the Administration Console, click *Access Manager > Auditing.*
- **Identity Server:** In the Administration Console, click *Access Manager > Identity Servers > Configuration Assignment > Logging.*
- **Access Gateway:** In the Administration Console, click *Access Manager > Access Gateways > Edit > Novell Audit.*

- **J2EE Agent:** In the Administration Console, click *Access Manager > J2EE Agents > Edit*.
- **SSL VPN:** In the Administration Console, click *Access Manager > SSL VPNs > Edit > Novell Audit Settings*.

In addition to the selectable events, device-generated alerts are automatically sent to the audit server. These Management Communication Channel events have an ID of 002e0605. All Access Manager events begin with 002e. SSL VPN starts with 0031. You can set up Novell Auditing to send e-mail whenever these events or your selected audit events occur. See Configuring System Channels (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/al6t4sd.html) in the Novell Audit 2.0 (http://www.novell.com/documentation/novellaudit20/treetitl.html) guide.

For information about audit event IDs and field data, see Appendix G, "Access Manager Audit Events and Data," on page 721.

The Access Gateway also supports a syslog and allows you to send e-mail notification to system administrators. To configure this system:

- **Linux Access Gateway:** In the Administration Console, click *Access Manager > Access Gateways > Edit > Alerts.*
- **NetWare Access Gateway:** In the Administration Console, click *Access Manager > Access Gateways > Edit > Legacy Alerts.*

1.5 Identity Server

By default, all Access Manager components (Identity Server, Access Gateway, SSL VPN, and J2EE* agents) trust the certificates signed by the local CA. We recommend that you configure the Identity Server to use an SSL certificate signed externally, and that you configure the trusted store of the service provider for each component to trust this new CA. See Chapter 25, "Assigning Certificates to Access Manager Devices," on page 375.

- Section 1.5.1, "Federation Concerns," on page 27
- Section 1.5.2, "Authentication Contracts," on page 27

1.5.1 Federation Concerns

When you set up federation between an identity provider and a service provider, you can select either to exchange assertions with a post method or to exchange artifacts. An artifact is a randomly generated ID, it contains no sensitive data, and only the intended receiver can use it to retrieve assertion data. Assertions might contain the user's password or other sensitive data, which can make them less secure than an artifact when the assertion is sent to the browser. It is possible for a virus on the browser machine to access the memory where the browser decrypts the assertion. If both providers support artifacts, you should select this method because it is more secure. For more details, see the Response Protocol Binding option in Section 10.1, "Configuring Authentication for a Trusted Identity Provider," on page 157.

1.5.2 Authentication Contracts

By default, the Administration Console allows you to select from the following contracts and options when specifying whether a resource requires an authentication contract:

- **None:** Allows public access to the resource and does not require authentication contract.

- **Name/Password - Basic:** Requires that the user enter a name and password that matches an entry in an LDAP user store. The credentials do not need to be sent over a secure port. This uses the unprotected BasicClass, which is not recommended for a production environment.
- **Name/Password - Form:** Requires that the user enter a name and password that matches an entry in an LDAP user store. The credentials do not need to be sent over a secure port. This uses the unprotected PasswordClass, which is not recommended for a production environment
- **Secure Name/Password - Basic:** Requires that the user enter the name and password from a secure (SSL) connection. This uses the ProtectedBasicClass, which is recommended for a production environment. If your Web servers are using basic authentication, this contract provides the credentials for this type of authentication.
- **Secure Name/Password - Form:** Requires that the user enter the name and password from a secure (SSL) connection. This uses the ProtectedPasswordClass, which is recommended for a production environment.
- **Any Contract:** Allows the user to use any contract defined for the Identity Server configuration.

If you have set up the Access Manager to require SSL connections among all of its components, you should delete the Name/Password - Form and the Name/Password - Basic contracts. This removes them from the list of available contracts when configuring protected resources and prevents them from being assigned as the contract for a protected resource. If these contracts are assigned, the user's password goes across the wire in clear text format. At some future date, if your system needs this type of contract, you can re-create it from the method. To delete these contracts, go to the Administration Console and click *Identity Servers* > *Servers* > *Edit* > *Local* > *Contracts*.

1.6 NetWare Access Gateway

The NetWare® Access Gateway is installed with two user accounts: config and admin. The config user has no assigned password and the admin user is given the password of novell.

IMPORTANT: Before your Access Gateways is placed in a production environment, you need to assign a password for the config user, and you need to change the password for the admin user. For instructions, see Section 15.6.3, "Setting the Password for the admin and config Users," on page 263.

Intruder detection lockout has been set up for these accounts. The config and admin users are allowed 5 attempts to log in successfully. If the user fails on the fifth attempt, the account is locked for 15 minutes.

Before you enable any of the following protocols, you need to be aware of their security issues:

- **Telnet:** Opens a clear text communication channel and sends passwords in clear text.
- **FTP:** Opens a clear text communication channel and sends passwords in clear text.
- **SSH:** Requires a LDAPS listener on port 636, on all IP addresses configured for the NetWare Access Gateway. It cannot be restricted to a single IP address.
- **SFTP:** Requires the `NCPIP.NLM` to be loaded with a listener on port 524.

If you enable any of these protocols, the NetWare Access Gateway needs to be installed behind a firewall appliance, and the firewall needs to block the following ports:

- 21 for FTP

- 23 for Telnet
- 524 for SFTP
- 636 for SSH

For more information about installing the Access Gateway behind a firewall, see "Setting Up Firewalls" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

1.7 Linux Access Gateway

You need to develop a security policy for the following:

- Section 1.7.1, "Default User Accounts," on page 29
- Section 1.7.2, "The SSH Protocol," on page 29
- Section 1.7.3, "The Via Header," on page 30

1.7.1 Default User Accounts

The Linux Access Gateway is installed with two user accounts: `config` and `root` with the password as `novell`. The *Novell Access Manager 3.0 SP3 IR2 Installation Guide* provides the following instructions for changes the passwords. Before your Access Gateway is placed in a production environment, make sure you have complete them.

1 Log in as `root` and change the password.

  1a At the login prompt, enter `root`.

  1b At the password prompt, enter `novell`.

  1c To change the password, enter `passwd`.

  1d Enter a password.

  1e Confirm the password by entering it again.

2 To change the password for the `config` user, enter the following commands:

  2a Enter `passwd config`.

  2b Enter a new password.

  2c Confirm the password by entering it again.

1.7.2 The SSH Protocol

Before you enable the SSH protocol, it requires an LDAPS listener on port 636, on all IP addresses configured for the Linux Access Gateway. It cannot be restricted to a single IP address:

If SSH is enabled, the Linux Access Gateway needs to be installed behind a firewall appliance, and the firewall needs to block port 636 for SSH.

For more information about installing the Access Gateway behind a firewall, see "Setting Up Firewalls" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

1.7.3 The Via Header

By default, the Via header is enabled and sent with requests. The Via header contains the version and build number of the Linux Access Gateway. If you have enabled telnet, this version information is available from a telnet command. If your security policy considers this a security risk, you need to disable the Via header.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxies/Authentication*.
2. In the *Embedded Service Provider* section, make sure that the *Enable Via Header* option is not selected.

   This is a global option that affects all defined reverse proxies and proxy services.
3. Click *OK* twice, then update the Access Gateway.

1.8 SSL VPN

For the best security properties, using the product in Enterprise mode is recommended. You should also install the client software prior to first use. For more information, see "Accessing SSL VPN in Enterprise Mode" in the *Novell Access Manager 3.0 SP 3 SSL VPN User Guide*.

Before you enable the connection, examine the certificate of the server that is asking for the authentication credentials. In order to prevent the phishing attacks, avoid connecting to a non-trusted server during the authentication phase.

Pre-installation of kernel drivers is recommended because of security concerns about installing non-trusted software.

In Enterprise mode, the tunnel is established between the client and server machines. This solution is not appropriate for multi-user machines where the users are logged in at the same time, because any software acting on behalf of an authenticated user on the client can make use of the encrypted tunnel.

Using AES 256 mode of encryption is recommended.

1.9 J2EE Agent

All communication should be sent over a secure channel.

Backing Up and Restoring Components

2

Backup and restore utilities are scripts that are run from the command line, and they allow you to back up and restore your Administration Console. An additional script allows you to export your configuration so Novell® Support can help diagnose possible configuration problems.

IMPORTANT: You cannot restore data from a previous version of Access Manager to a new version. It is recommended that you create a new configuration backup whenever you upgrade to a newer version of Access Manager.

Before running these scripts, verify the following:

- You have `root` access.
- You have changed the directory to `/opt/novell/devman/bin`.

This section discusses the following topics:

- Section 2.1, "How The Backup and Restore Process Works," on page 31
- Section 2.2, "Backing up the Administration Console," on page 32
- Section 2.3, "Restoring an Administration Console," on page 32
- Section 2.4, "Restoring an Identity Server," on page 34
- Section 2.5, "Restoring an Access Gateway," on page 34
- Section 2.6, "Running the Diagnostic Configuration Export," on page 35

2.1 How The Backup and Restore Process Works

- "Default Parameters" on page 31
- "Certtool.jar" on page 31

Default Parameters

All of the scripts call the `getparams.sh` script to request the parameters from the user. The `defbkparm.sh` script is created by the Access Manager installation. It contains default parameters for several of options required by the underlying backup and restore utilities. If the entries in this file are commented out, the user is prompted for the additional parameters.

Certtool.jar

`Certtool.jar` is a key certificate utility for eDirectory™ built on top of the same interfaces that the Access Manager certificate management features use. It provides some features similar to the Java* keytool utility. It must run on a computer that has eDirectory installed, or at least NPKI. The basic command line to invoke the tool is:

```
java -Djava.library.path=/opt/novell/lib -jar certtool.jar -h
```

The -h option produces help listing of command line options.

2.2 Backing up the Administration Console

1. Change to the `/opt/novell/devman/bin` directory.
2. Run the following command from `root`: `./ambkup.sh`.
3. Enter the Access Manager administration user ID.
4. Enter the Access Manager administration password.
5. Re-enter the password for verification.
6. Enter a password for encrypting and decrypting private keys, then re-enter for verification.

   You must use the same password for both backup and restore.
7. Press Enter.

The backup script creates a `.zip` file containing several files, including the certificate information. This `.jar` file uses the same API that Access Manager's certificate management features use. The `.zip` file contains the following:

- The configurations store's CA key.
- The certificates contained in the configuration store.
- The trusted roots in the trustedRoots container in accessManagerContainer. The trusted roots are backed up in both the LDIF file and the ZIP file. They are added to the ZIP file so that the ZIP file has the complete certificate-related configuration.
- An LDIF file. This file is created by the eDirectory ICE utility. Everything in the OU=accessManagerContainer,O=novell container is exported to the LDIF file.

IMPORTANT: The backup utility prompts you for a location to store the backup file, so that it is not erased if you uninstall the product. The default location is the logged-in user's home directory.

2.3 Restoring an Administration Console

The restore script replaces the configuration records in the configuration database with the records in the backup of the configuration store. The restore script should not be used to move configuration data from one machine to another. It should be used to restore configuration data to a machine that has failed.

The restoration steps are dependent upon whether the Administration Console is installed on its own machine or with an Identity Server:

- Section 2.3.1, "Restoring a Standalone Administration Console," on page 32
- Section 2.3.2, "Restoring an Administration Console with an Identity Server on the Same Machine," on page 33

2.3.1 Restoring a Standalone Administration Console

If you are performing a restore because the Administration Console machine failed, install the same version of the Administration Console on the new machine. Then perform the restore.

1. Ensure that the `.zip` file created during the backup process is accessible.

2. Change to the `/opt/novell/devman/bin` directory.
3. Run the following command from `root: ./amrestore.sh`.
4. Enter the Access Manager administration user ID.
5. Enter the Access Manager administration password.
6. Enter the name of the backup file. Do not include the `.zip extension.`
7. Enter the private key encryption password and press Enter.
8. Re-enter the private key encryption password and press Enter.

If you have secondary Administration Console installed, you must restart Tomcat (`/etc/init.d/novell-tomcat4 restart`) in order to re-establish LDAP connections to the primary Administration Console.

If you are restoring only the Administration Console, other components should still function properly after the restore.

2.3.2 Restoring an Administration Console with an Identity Server on the Same Machine

If you are performing a restore because the machine failed, install the same version of the Administration Console on the new machine. Do not reinstall the Identity Server at this time. The following procedures explain when the Identity Server should be reinstalled.

IMPORTANT: Whenever you run the `amrestore.sh` script, the Administration Console is restored as a standalone Administration Console. You must perform the steps described in Step 9 to restore your Identity Server into the configuration.

---

1. Ensure that the `.zip` file created during the backup process is accessible.
2. Change to the `/opt/novell/devman/bin` directory.
3. Run the following command from `root: ./amrestore.sh`.
4. Enter the Access Manager administration user ID.
5. Enter the Access Manager administration password.
6. Enter the name of the backup file. Do not include the `.zip extension.`
7. Enter the private key encryption password and press Enter.
8. Re-enter the private key encryption password and press Enter.
9. For the Identity Server, complete the following steps after the restore has finished:
    - **9a** Remove the Identity Server from the cluster configuration. (See Section 6.1.3, “Removing a Server from a Configuration,” on page 60.)
    - **9b** Delete the Identity Server from the Administration Console. (See Section 5.1, “Managing an Identity Server,” on page 49.)
    - **9c** Uninstall the Identity Server. (See “Uninstalling the Identity Server” in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

      This is required if the Identity Server is installed on the machine. If you installed the Identity Server before running the `amrestore.sh` script, you need to uninstall the Identity Server.

  - **9d** Install the Identity Server. (See "Installing the Novell Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
  - **9e** Reassign the Identity Server to the cluster configuration that it was removed from. (See Section 6.1.2, "Assigning an Identity Server to a Cluster Configuration," on page 60.)

If you have secondary Administration Console installed, you must restart Tomcat (`/etc/init.d/novell-tomcat4 restart`) in order to re-establish LDAP connections to the primary Administration Console.

2.4 Restoring an Identity Server

1. Remove the Identity Server from the Identity Server cluster configuration. (See Section 6.1.3, "Removing a Server from a Configuration," on page 60.)
2. Delete the Identity Server from the Administration Console. (See Section 5.1, "Managing an Identity Server," on page 49.)
3. Uninstall the Identity Server. (See "Uninstalling the Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

   This might not be necessary, if you used a new machine for the restoring the configuration.
4. Install the new Identity Server, which imports it into the Administration Console. (See "Installing the Novell Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
5. Assign the new server to the Identity Server cluster configuration. (See Section 6.1.2, "Assigning an Identity Server to a Cluster Configuration," on page 60.)

2.5 Restoring an Access Gateway

If an Access Gateway machine experiences a hardware failure, such as a failed hard disk, you can preserve its configuration and have it applied to the replacement machine.

- Section 2.5.1, "Clustered Access Gateway," on page 34
- Section 2.5.2, "Single Access Gateway," on page 35

2.5.1 Clustered Access Gateway

If the hardware fails on an Access Gateway machine that belongs to a cluster, do the following:

1. In the Administration Console, view the configuration of the cluster. Click *Access Manager* > *Access Gateways*.
2. (Conditional) If the failed Access Gateway is the primary server, assign another server to be the primary server.
   - **2a** On the Access Gateways page, click *[Name of Cluster]* > *Edit*.
   - **2b** For the *Primary Server* field, select another server to be the primary server, then click *OK* > *Close*.
   - **2c** Click *Identity Servers* > *Update*.
3. Delete the failed Access Gateway from the cluster. Click *Access Gateways*, select the failed Access Gateway, then click *Actions* > *Remove from Cluster*.

IMPORTANT: Do not delete the Access Gateway from the Administration Console.

---

4. On the new machine, install the Access Gateway, specifying the same Administration Console, IP address, host name, and domain name as the failed machine.
5. When the machine imports into the Administration Console, add the machine to the Access Gateway cluster.
   - **5a** In the Administration Console, click *Access Manager* > *Access Gateways*.
   - **5b** Select the name of the Access Gateway, then click *Actions* > *Assign to Cluster* > *[Name of Cluster]*.

2.5.2 Single Access Gateway

If the failed Access Gateway is a single machine and you want to preserve its configuration, you need to perform the following steps:

1. Do not delete the Access Gateway from the Administration Console.

   If you delete the Access Gateway from the Administration Console, the configuration information is deleted.
2. On the new machine, install the Access Gateway software, using the same IP address, host name, and domain name as the failed device and specifying the same Administration Console.
3. When the installation has completed and the device has been imported in the Administration Console, verify the following:
   - **3a** Check its trusted relationship with the Identity Server. Click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxy / Authentication*.
   - **3b** (Linux Access Gateway only) If you have configured the Access Gateway to use SSL, reconfigure the certificates for the listener. Click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.
   - **3c** Save and apply any changes.

2.6 Running the Diagnostic Configuration Export

To create an `.ldif` file that you can export for diagnostic purposes:

1. Change to the `/opt/novell/devman/bin` directory.
2. Run the following command from `root`: `./amdiagcfg.sh`.
3. Enter the Access Manager administration user ID.
4. Enter the Access Manager password.
5. Re-enter the password for verification.
6. Press Enter.

The diagnostic configuration export utility is almost identical to the backup utility with two differences: the ZIP file is not created, and the final LDIF file is scanned to have passwords removed. Passwords are blanked out by a program called Strippasswd.

Strippasswd removes occurrences of passwords in the LDIF file, replacing them with empty strings. If you look at the LDIF file, you will see that password strings are blank. You might see occurrences within the file or text that looks similar to password="String". These are not instances of passwords, but rather definitions that describe passwords as string types.

The LDIF file can then be sent to Novell Support for help in diagnosing configuration problems.

Administration Console

3

This section discusses the following Administration Console topics:

- Section 3.1, “Administration Console Conventions,” on page 37
- Section 3.2, “Starting and Stopping Access Manager Components,” on page 37
- Section 3.3, “Changing the Password for the Administration Console,” on page 42
- Section 3.4, “Multiple Administrators, Multiple Sessions,” on page 42

For information about installing secondary consoles for fault tolerance, see “Clustering and Fault Tolerance” in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

For troubleshooting information about converting a secondary console into a primary console, see Section 37.5, “Converting a Secondary Console into a Primary Console,” on page 580.

3.1 Administration Console Conventions

- The required fields on a configuration page contain an asterisk by the field name.
- All actions such as delete, stop, and purge require verification before they are executed.
- Changes are not applied to a server until you update the server.
- Sessions are monitored for activity. If your session becomes inactive, you are asked to log in again and unsaved changes are lost.
- Right-clicking links in the interface, then selecting to open the link in a new tab or window is not supported. If the left navigation panel is not present in the window or tab, close the session and start a new one.

3.2 Starting and Stopping Access Manager Components

Access Manager has three services that can be stopped and started: the Identity Server, the Access Gateway, and the embedded service provider within the Access Gateway. Normally, you do not need to stop and start these services. However, if you need to change certain configuration options, you can be prompted to update the Identity Server or to restart the embedded service provider.

The following sections explain how to update, stop, start, and schedule a restart of the various Access Manager components:

- Section 3.2.1, “Updating an Identity Server Configuration,” on page 38
- Section 3.2.2, “Restarting the Identity Server,” on page 39
- Section 3.2.3, “Updating the Access Gateway,” on page 39
- Section 3.2.4, “Restarting the Access Gateway Service Provider,” on page 40
- Section 3.2.5, “Starting the Access Gateway Service Provider,” on page 40
- Section 3.2.6, “Stopping the Access Gateway Service Provider,” on page 40
- Section 3.2.7, “Rebooting the Access Gateway,” on page 40

- Section 3.2.8, "Scheduling a Reboot of the Access Gateway," on page 41
- Section 3.2.9, "Stopping the Access Gateway," on page 41
- Section 3.2.10, "Scheduling the Shutdown of the Access Gateway," on page 41

3.2.1 Updating an Identity Server Configuration

Whenever you change an Identity Server configuration, the system prompts you to update the configuration. An *Update Servers* status is displayed under the *Status* column on the Servers page. You must click *Update Servers* to update the configuration so that your changes take effect.

When clicked, this link sends a reconfigure command to all servers that use the configuration. The servers then begin the reconfiguration process. This process occurs without interruption of service to users who are currently logged in.

When you update a configuration, the system blocks inbound requests until the update is complete. The server checks for any current requests being processed. If there are such requests in process, the server waits five seconds and tests again. This process is repeated three times, thus waiting up to fifteen seconds for these requests to be serviced and cleared out. After this period of time, the update process begins. Any remaining requests might have errors.

During the update process, all settings are reloaded with the exception of the base URL. In most cases, user authentications are preserved; however, there are conditions during which some sessions are automatically timed out. These conditions are:

- A user logged in via an authentication contract that is no longer valid. This occurs if an administrator removes a contract or changes the URI that is used to identify it.
- A user logged in to a user store that is no longer valid. This occurs if you remove a user store or change its type. Changing the LDAP address to a different directory is not recommended, because the system does not detect the change.
- A user received authentication from an identity provider that is no longer trusted. This occurs if you remove a trusted identity provider or if the metadata for the provider changed.

Additionally, if you remove a service provider from an identity provider, the identity provider removes the provided authentication to that service provider. This does not cause a timeout of the session to occur.

Changes to the SAML and Liberty protocol profiles can result in the trusted provider having outdated metadata for the Identity Server being reconfigured. This necessitates an update at the other provider and might cause unexpected behavior until that occurs.

1. In the Administration Console, click *Access Manager* > *Identity Servers*, then click the *Servers* tab.
2. Select the Identity Server configuration, then click *Update Servers*.

   This link is available only when you have made changes that require a server update.

3.2.2 Restarting the Identity Server

Starting and stopping an Identity Server terminates active user sessions. These users receive a prompt to log in again.

1. In the Administration Console, click *Access Manager* > *Identity Servers* and select the Identity Server to stop.
2. Click *Stop*.
3. Wait for the *Command Status* to change from *Pending* to *Complete*.
4. Select the Identity Server, then click *Start*.
5. When the *Command Status* changes to *Complete*, click *Refresh*.

   The status icon of the Identity Server should turn green.

3.2.3 Updating the Access Gateway

When a configuration change has been made, but not applied, the Access Gateway is in an *Update* status on the Access Gateways page. If the Access Gateway is a member of a cluster, the cluster is in an *Update All* status. You can click *Update* to apply the configuration change to a single Access Gateway or *Update All* to apply the configuration change to all members of a cluster.

If the changes have been saved to browser cache, but not to the configuration store, the changes are lost if your session times out before you apply the changes. The Access Gateway remains in an *Update* status, but when you click *Update*, there are no changes to apply. If you prefer to update members of a cluster one at a time, it is best to save the changes to the configuration datastore before applying them. Click *Edit*, then click *Save*.

When you click *Update*, three options are displayed:

- When you have modified services of the Access Gateway, the update option for *All Configuration* is available. Depending upon what has been modified, updating might cause logged in users to lose data and their connections.
- When the ESP logging settings have been modified on the Identity Server, the update option for *Logging Settings* is available. The *Logging Settings* option causes no interruption in services.
- If a policy is modified that the server has enabled for a protected resource or a protected resource has a policy enabled or disabled and the policy changes are the only modifications that have occurred, the update option for *Policy Settings* is available. The Policy Settings option causes no interruption in services.

When you make the following configuration changes, the *Update All* option is the only option available and your site will be unavailable while the update occurs:

- The Identity Server configuration that is used for authentication is changed (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication,* then select a different value for the *Identity Server Cluster* option).
- A different reverse proxy is selected to be used for authentication (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication*, then select a different value for the *Reverse Proxy* option).
- The protocol or port of the authenticating reverse proxy is modified (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication* > *[Name of Reverse Proxy]*, then change the SSL options or the port options).

- The published DNS name of the authentication proxy service is modified (*Access Gateways > Edit > Reverse Proxy/Authentication > [Name of Reverse Proxy] > [Name of First Proxy Service]*, then modify the *Published DNS Name* option).

3.2.4 Restarting the Access Gateway Service Provider

To stop and start the Access Gateway service provider:

1. In the Administration Console, click *Access Manager* > *Access Gateways*, then select the Access Gateway, then click *Actions*.
2. Click *Service Provider* > *Restart Service Provider*, then click *OK*.

   In a few seconds, the *Health* icon of the Access Gateway should turn green.

3.2.5 Starting the Access Gateway Service Provider

When an Access Gateway is removed from a cluster configuration, the embedded service provider is stopped. It should remain stopped until you have reconfigured the Access Gateway. When you have finished the reconfiguration, you should start the embedded service provider.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, then select the Access Gateway, then click *Actions*.
2. Click *Service Provider* > *Start Service Provider*, then click *OK*.

   In a few seconds, the Health icon of the Access Gateway should turn green.

3.2.6 Stopping the Access Gateway Service Provider

Stopping the embedded service provider is a quick way to make the Access Gateway inaccessible to users.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, then select the Access Gateway, then click *Actions*.
2. Click *Service Provider* > *Stop Service Provider*, then click *OK*.

   In a few seconds, the status icon of the Access Gateway should turn red.

3.2.7 Rebooting the Access Gateway

Rebooting the Access Gateway makes all protected resources unavailable until the Access Gateway returns to a server status of green. The Access Gateway is stopped, and the operating system is rebooted.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, then select the Access Gateway.
2. Click *Reboot*.

   In a few minutes, the status icon of the Access Gateway should turn green.

3.2.8 Scheduling a Reboot of the Access Gateway

Rebooting the Access Gateway makes all protected resources unavailable until the Access Gateway returns to a server status of green. Scheduling this event allows you to pick the best time for your resources to be momentarily unavailable.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, select the Access Gateway, then click *Actions*.
2. Click *Schedule Reboot*.

   The following field displays information about the command you are scheduling.

   Type: Displays the type of command that is being scheduled, such as *Access Gateway Shutdown, Access Gateway Reboot, Access Gateway Upgrade, Device Configuration*.
3. Fill in the following fields:

   Name Scheduled Command: (Required) Specifies a name for this scheduled command. This name is used in log and trace files.

   Description: (Optional) Provides a field to describe the reason for the command.

   Date & Time: The drop-down menus allow you to select the day, month, year, hour, and minute when the command should execute.
4. Click *OK*.

3.2.9 Stopping the Access Gateway

You should stop the Access Gateway only when you plan to turn off the power or to configure boot options for troubleshooting. After you have stopped the Access Gateway, you must have physical access to the machine to start it.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, select the Access Gateway, then click *Shutdown*.
2. To confirm the shutdown, click *OK*.

The machine is physically turned off. Before you start the Access Gateway again, you can modify the boot options on a NetWare Access Gateway. For information about these boot options, see Section 40.3.1, "Additional Options During the Boot Process," on page 657.

3.2.10 Scheduling the Shutdown of the Access Gateway

You should stop the Access Gateway only when you plan to turn off the power or to configure boot options for troubleshooting. After you have stopped the Access Gateway, you must have physical access to the machine to start it. Scheduling this event allows you to pick the best time for the Access Gateway to be unavailable.

1. In the Administration Console, click *Access Manager* > *Access Gateways*, select the Access Gateway, then click *Actions*.
2. Click *Schedule Shutdown*.

   The following field displays information about the command you are scheduling.

   Type: Displays the type of command that is being scheduled, such as *Access Gateway Shutdown, Access Gateway Restart, Access Gateway Upgrade, Device Configuration*.

3. Fill in the following fields:

   Name Scheduled Command: (Required) Specifies a name for this scheduled command. This name is used in log and trace files.

   Description: (Optional) Provides a field to describe the reason for the command.

   Date & Time: The drop-down menus allow you to select the day, month, year, hour, and minute when the command should execute.

4. Click *OK*.

   The machine is turned off when the scheduled command executes.

Before you start the Access Gateway again, you can modify the boot options on a NetWare Access Gateway. For information about these boot options, see Section 40.3.1, "Additional Options During the Boot Process," on page 657.

3.3 Changing the Password for the Administration Console

The admin of the Administration Console is a user created in the novell container of the configuration store. To change the password:

1. In the Administration Console, click *Users* > *Modify User*.
2. Click the *Object Selector* icon.
3. Browse the novell container and select the name of the admin user, then click *OK*.
4. Click *Restrictions* > *Set Password*.
5. Enter a password in the *New password* text box.
6. Confirm the password in the *Retype new password* text box.
7. Click *OK* twice.

3.4 Multiple Administrators, Multiple Sessions

The Administration Console has been designed to warn you when another administrator is making changes to a policy container or to an Access Manager device (such as an Access Gateway, SSL VPN, or J2EE Agent). The person who is currently editing the configuration is listed at the top of the page with an option to unlock and with the person's distinguished name and IP address. If you select to unlock, you destroy all changes the other administrator is currently working on.

WARNING: Currently, locking has not been implemented on the pages for modifying the Identity Server. If you have multiple administrators, they need to coordinate with each other so that only one administrator is modifying an Identity Server cluster at any given time.

Multiple Sessions: You should not start multiple sessions to the Administration Console with the same browser on a workstation. Browser sessions share settings that can result in problems when you apply changes to configuration settings. However, if you are using two different brands of browsers simultaneously, such as Internet Explorer* and Firefox*, it is possible to avoid the session conflicts.

Multiple Admin Accounts: The Administration Console is installed with one admin user account. If you have multiple administrators, you might want to create a user account for each one so that log files reflect the modifications of each administrator. The easiest way to do this is to create an account for each administrator and make the user security equivalent to the admin user.

1. In the Administration Console, click *Users* > *Create User*.
2. Create a user account for each administrator.
3. Click *Modify User*, then select the created user.
4. Click *Security* > *Security Equal To*.
5. Select the admin user, then click *Apply* > *OK*.
6. Repeat Step 3 through Step 5 for each user you want to make security equivalent to the admin user.

Changing the IP Address of Access Manager Devices

4

The following sections explain how to change the IP address on the following devices:

- Section 4.1, “Changing the IP Address of the Administration Console,” on page 45
- Section 4.2, “Changing the IP Address of an Identity Server,” on page 45
- Section 4.3, “Changing the IP Address of the Access Gateway,” on page 46
- Section 4.4, “Changing the IP Address of an Audit Server,” on page 47

NOTE: Changing the IP address of an SSL VPN component is not recommended.

4.1 Changing the IP Address of the Administration Console

We recommend that you install the Administration Console with the IP address that it will always use because all of the devices that import into the Administration Console use this address to establish secure communication with the Administration Console.

The only tested method of changing the IP address so that all other devices trust the Administration Console is to install a secondary console with the new IP address and then promote the secondary console to be the primary console. Remember to change the IP addresses of all components pointing to the new Administration Console.

See the following sections:

- “Installing Secondary Versions of the Administration Console” in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*
- “Converting a Secondary Console into a Primary Console” on page 580

Converting a secondary console into a primary console is not a simple task. The task was designed as a disaster recovery solution when the primary console is no longer available. It is not a simple configuration change.

4.2 Changing the IP Address of an Identity Server

These instructions assume that your Identity Server and Administration Console are not on the same machine. If they are on the same machine, see Section 4.1, “Changing the IP Address of the Administration Console,” on page 45.

To move a machine or change the IP address for the Identity Server:

1. In the Administration Console, click *Access Manager* > *Identity Servers*.
2. Click the server name.
3. On the General page, click *Edit*.

4 Specify the new IP address in the *Management IP Address* field and, if necessary, a port.

5 Click *OK*, then click *Close*.

6 In Linux, open the console shell and stop the server communication service by using the following command:

```
/etc/init.d/novell-jcc stop
```

7 Using YaST, change the IP address on the physical Linux server hosting the Identity Server.

8 At the console shell, access the `/opt/novell/devman/jcc` directory, then enter the following command:

```
conf/Configure.sh
```

9 When you are prompted for the local listener IP address, enter the new IP.

10 When you are prompted for the administration server IP, enter the same IP that you used during the initial installation.

11 Follow the prompts and accept the defaults for ports and admin user.

12 At the console shell, start the server communication service using the command:

```
/etc/init.d/novell-jcc start
```

13 Restart the Identity Server application on the Servers page.

For information about deleting an Identity Server, see Section 5.1, "Managing an Identity Server," on page 49.

4.3 Changing the IP Address of the Access Gateway

If you need to change the IP address of the Access Gateway machine, you need to configure the Access Gateway for this change. This is especially significant when the Access Gateway machine has only one IP address.

IMPORTANT: The new IP address must be configured in the Administration Console before you change it on the Access Gateway. If you change in on the Access Gateway first, the Administration Console will not trust the Access Gateway and will not establish communication.

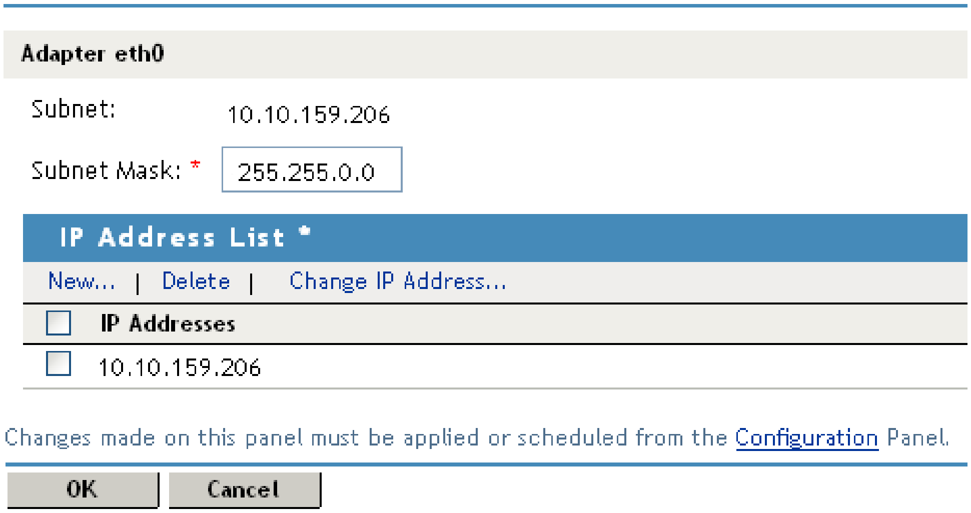
---

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Adapter List*.

2 (Conditional) If the machine belongs to a cluster, select the Access Gateway from the *Cluster Member* list.

3 From the Adapter List, select the subnet mask that contains the IP address you want to change.

When you select the subnet mask, the Adapter page appears.


Adapter eth0
Subnet: 10.10.159.206
Subnet Mask: * 255.255.0.0
IP Address List *
New... | Delete | Change IP Address...
☐ IP Addresses
☐ 10.10.159.206
Changes made on this panel must be applied or scheduled from the Configuration Panel.
OK
Cancel

4 Select the old IP address, click *Change IP Address*, specify the new IP address, then click *OK*.

This option changes all configuration instances of the old IP address to the new IP address. For example, any reverse proxies that have been assigned the old IP address as a listening address are modified to use the new IP address as the listening address.

5 To save your changes to browser cache, click *OK*.

6 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

7 If you are physically moving the machine, move it before completing the rest of these steps.

8 Check the IP address that the Administration Console uses for managing the Access Gateway. Click *Access Gateways* > *[Name of Access Gateway]* > *Edit*.

9 If the old IP address is listed as the *Management IP Address*, select the new IP address. If your Access Gateway has multiple IP addresses, select the one that you want the Administration Console to use for communication with the Access Gateway.

The port should only be modified if there is another device on the Access Gateway that is using the default port of 1443.

10 If the name of the Access Gateway is the old IP address, modify the *Name* option.

11 Click *OK*.

The Administration Console uses the configured IP address to find the Access Gateway.

If your Access Gateway stops reporting to the Administration Console after completing these steps, you need to trigger an auto-import. See “Triggering an Import Retry” in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

4.4 Changing the IP Address of an Audit Server

To move a machine or change the IP address for the audit server:

1 In the Administration Console, click *Access Manager* > *Auditing*.

2 On the Novell Auditing page, change the IP address for the server and, if necessary, the port.

3 Click *OK*.

4 Update all Access Gateways.

5 Reboot all servers to use the new configuration.

Maintaining an Identity Server

5

Server maintenance involves tasks that you perform after you have configured the server. Maintenance includes monitoring server and statistics, configuring Identity Server logging, replacing certificates, and so on.

- Section 5.1, "Managing an Identity Server," on page 49
- Section 5.2, "Editing Server Details," on page 51

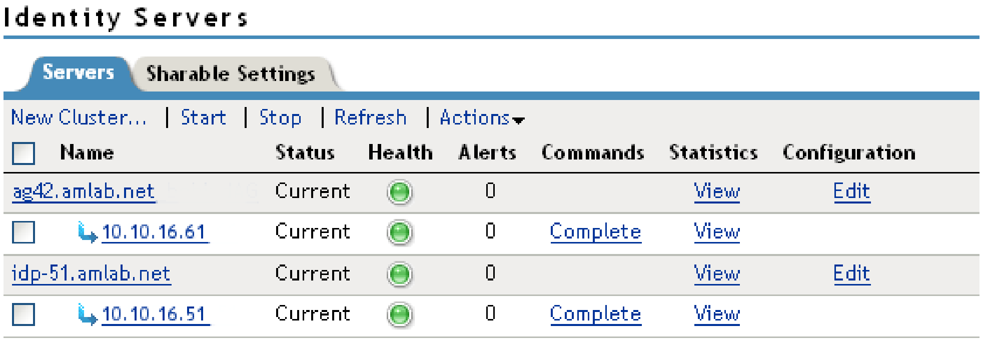
For information about server health, see Section 34.2, "Monitoring the Health of an Identity Server," on page 552.

For information about configuring the Identity Server, see Part II, "Novell Identity Server Configuration," on page 53.

5.1 Managing an Identity Server

The Identity Servers page is the starting point for managing Identity Servers. Most often, you use this page to stop and start servers, and to assign servers to Identity Server configurations. An Identity Server cannot operate until you have assigned it to an Identity Server configuration.

1 In the Administration Console, click *Access Manager* > *Identity Servers*.



☐ Name	Status	Health	Alerts	Commands	Statistics	Configuration
ag42.amlab.net	Current	●	0		View	Edit
☐ 10.10.16.61	Current	●	0	Complete	View	
idp-51.amlab.net	Current	●	0		View	Edit
☐ 10.10.16.51	Current	●	0	Complete	View	

2 On the *Servers* tab, you can perform the following functions by clicking the server's check box, then clicking any of the following options:

New Cluster: Creates a new cluster configuration. See Section 6.1.1, "Creating a Cluster Configuration," on page 56.

Start: Starts the selected server. (See Section 3.2, "Starting and Stopping Access Manager Components," on page 37.)

Stop: Stops the selected server.

Refresh: Refreshes the server list.

Actions: Enables you to perform the following tasks:

- **Assign to Cluster:** Enables you to assign a server to a cluster configuration. See Section 6.1.2, "Assigning an Identity Server to a Cluster Configuration," on page 60 for more information.

- **Remove from Cluster:** Enables you to remove one or more servers from a configuration. See Section 6.1.3, “Removing a Server from a Configuration,” on page 60 for more information.
- **Delete:** Deletes the selected server.

IMPORTANT: The system does not allow you to delete an Identity Server that is started. You must first stop the server, then delete it. This removes the configuration object from the configuration store on the Administration Console. To remove the server software from the machine where it was installed, you must run the uninstall script on the server machine.

- **Update Health from Server:** Performs a health check for the device.

This page also displays links in the following columns:

Column	Description
Name	Lists Identity Server and cluster configuration names.
Status	Lists the status of each configuration. **Current:** Indicates that the server is using the latest configuration data. If you change a configuration, the system displays an *Update* or *Update All* link. **Update:** A link to update an Identity Server’s configuration data without stopping the server. **Update All:** A link displayed for cluster configurations. This lets you update all the Identity Servers in a cluster to use the latest configuration data, with options to include logging and policy settings.
Health	Lists the health of each configuration and each server.
Alerts	Displays the Alerts page where you can monitor and acknowledge server alerts.
Commands	Displays the Command Status page.
Statistics	Displays the Server Statistics page and allows you to view the server statistics. See Section 33.1, “Monitoring Identity Server Statistics,” on page 535.
Configuration	Lists the Identity Server configuration to which this server belongs. An Identity Server can belong to multiple configurations.

5.2 Editing Server Details

You can edit server details, such as the server name and port. You can also access the other server management tabs from this page.

1. In the Administration Console, click *Access Manager* > *Identity Servers*, then click the server name.
2. Click *Edit*.
3. Fill in the following fields as necessary:

   Name: The name of the Identity Server. Names must be alphanumeric and can include spaces, hyphens, and underscores.

   Management IP Address: The IP address of the Identity Server. Changing server IP addresses is not recommended and causes the server to stop reporting. See Section 4.2, “Changing the IP Address of an Identity Server,” on page 45.

   Port: The Identity Server port.

   Location: The location of the Identity Server.

   Description: A description of the Identity Server.
4. To save your changes, click *OK*. Otherwise, click *Cancel*.

Novell Identity Server Configuration

II

In Access Manager, the Identity Server is responsible for authenticating users, building the user's role information, and distributing it to the various components. It also serves as the central point for components that request identity information.

This section of the Administration Guide describes the following topics:

- Chapter 6, "Configuring an Identity Server," on page 55
- Chapter 7, "Defining Shared Settings," on page 83
- Chapter 8, "Configuring Local Authentication," on page 89
- Chapter 9, "Configuring Trusted Providers," on page 139
- Chapter 10, "Configuring User Authentication and Federation," on page 157
- Chapter 11, "Configuring Communication Profiles," on page 171
- Chapter 12, "Configuring Liberty Web Services," on page 173

For conceptual information about Identity Server maintenance tasks, such as auditing, logging, and health monitoring, see Part VII, "Monitoring Access Manager Components," on page 503.

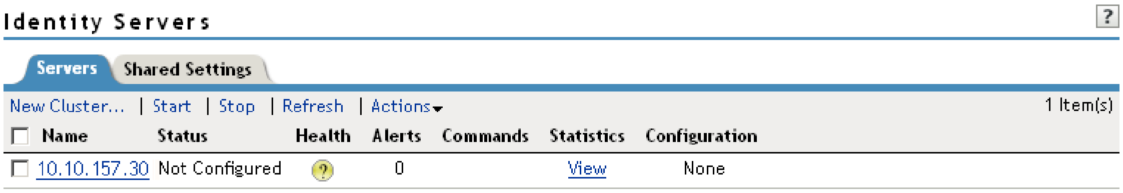
For conceptual information about Liberty and SAML, see the appropriate sections in Part IX, "Appendixes," on page 695.

This section of the administration guide is intended to help you understand and configure the Identity Server for authentication, and includes advanced topics. It is recommended that you first become familiar with the information in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*, which is intended to familiarize you with Access Manager and helps you understand how to perform a basic Identity Server configuration, cluster servers, set up a resource protected by an Access Gateway, and configure SSL. The Basic Setup and Administration guides are designed to work together, and important information and setup steps are not necessarily repeated in both places.

Configuring an Identity Server

6

After you log in to the Administration Console, click *Access Manager* > *Identity Servers*. The system displays the installed server.



Identity Servers

1 Item(s)

☐ Name	Status	Health	Alerts	Commands	Statistics	Configuration
☐ 10.10.157.30	Not Configured	?	0		View	None

At this point, the Identity Server is in an unconfigured state and is halted. It remains in this state and cannot function until you create an Identity Server configuration and assign the Identity Server to the new configuration. The configuration defines how the Identity Server functions in an Access Manager configuration. In an Identity Server cluster, multiple servers must use the same configuration.

- Section 6.1, "Managing a Cluster Configuration," on page 55
- Section 6.2, "Modifying the Base URL," on page 63
- Section 6.3, "Enabling Role-Based Access Control," on page 64
- Section 6.4, "Using netHSM for the Signing Key Pair," on page 64
- Section 6.5, "Configuring Secure Communication on the Identity Server," on page 78

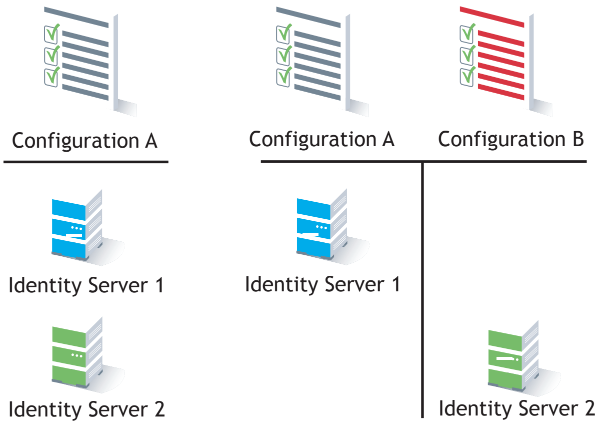
Additional Identity Server configuration topics for authentication include:

- Chapter 7, "Defining Shared Settings," on page 83
- Chapter 8, "Configuring Local Authentication," on page 89
- Chapter 12, "Configuring Liberty Web Services," on page 173

6.1 Managing a Cluster Configuration

After you install an Identity Server, you must create a cluster configuration in order to configure the Identity Server. You can assign the cluster configuration to one or more Identity Servers. As shown in Figure 6-1, you can also create multiple configurations and assign different Identity Servers to them.

Figure 6-1 Identity Server Configurations



When you assign multiple Identity Servers to the same configuration, you need to install an L4 switch, which allows the work load to be balanced among the machines.

Whether there is one machine or multiple machines in a cluster, the Access Manager software configuration process is the same. This section describes the following clustering tasks:

- Section 6.1.1, "Creating a Cluster Configuration," on page 56
- Section 6.1.2, "Assigning an Identity Server to a Cluster Configuration," on page 60
- Section 6.1.3, "Removing a Server from a Configuration," on page 60
- Section 6.1.4, "Managing a Cluster with Multiple Identity Servers," on page 60

6.1.1 Creating a Cluster Configuration

This section discusses the settings available for an Identity Server configuration, such as importing SSL certificates, enabling introductions, and configuring identity consumer settings. You should be familiar with "Creating a Basic Identity Server Configuration" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide* before proceeding.

An Identity Server always operates as an identity provider and can optionally be configured to run as an identity consumer (also known as a service provider), using Liberty, SAML 1.1, or SAML 2.0 protocols. In an Identity Server cluster, multiple servers use the same configuration.

In an Identity Server configuration, you specify the following information:

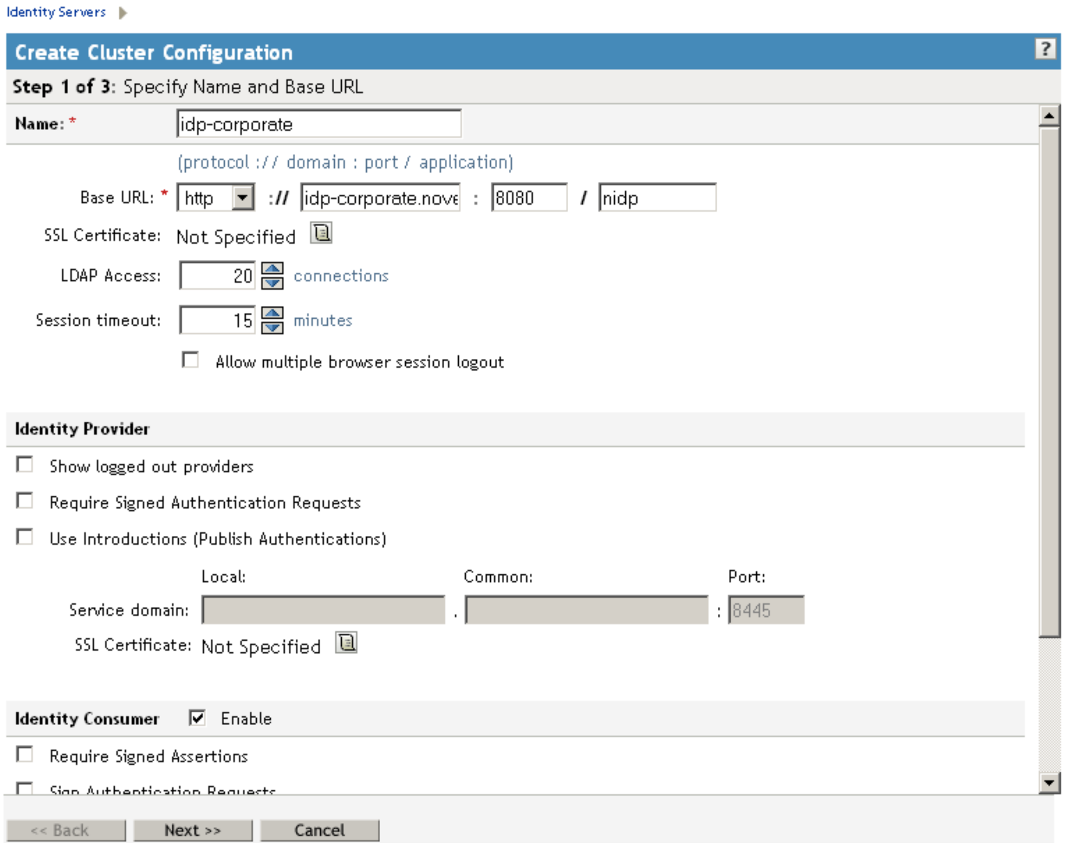
- The base URL for the server or clustered server site.
- Certificates for the Identity Server, identity provider, and identity consumer.
- Authentication settings, such as whether the identity provider requires signed authentications from service providers.
- The service domains used for publishing and discovering authentications.
- Organizational and contact information for the server, which is published in the metadata of the Liberty and SAML protocols.
- The LDAP directories (user stores) used to authenticate users, and the trusted root for secure communication between the Identity Server and the user store.

To create an Identity Server configuration:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers*.
2. Select the Identity Server's check box, then click *New Cluster*.

   Selecting the server is one way to assign it to the cluster configuration.
3. In the *New Cluster* dialog box, enter a name for the cluster configuration. If you did not select the server in the previous step, you can now select the server or servers that you want to assign to this configuration.

   For more information about assigning servers to a configuration, see Section 6.1.2, "Assigning an Identity Server to a Cluster Configuration," on page 60.
4. Click *OK*.



Identity Servers

Create Cluster Configuration

Step 1 of 3: Specify Name and Base URL

Name: * idp-corporate

(protocol :// domain : port / application)

Base URL: * http :// idp-corporate.nove : 8080 / nidp

SSL Certificate: Not Specified

LDAP Access: 20 connections

Session timeout: 15 minutes

☐ Allow multiple browser session logout

Identity Provider

☐ Show logged out providers

☐ Require Signed Authentication Requests

☐ Use Introductions (Publish Authentications)

Local: Common: Port:

Service domain: . : 8445

SSL Certificate: Not Specified

Identity Consumer ☑ Enable

☐ Require Signed Assertions

☐ Sign Authentication Requests

5. Fill in the following fields to specify the properties for your Identity Server configuration:

   Name: A name by which you want to refer to the configuration. This field is populated with the name you provided in the *New Cluster* dialog box. You can change this name here, if necessary.

   IMPORTANT: Carefully determine your settings for the base URL, protocol, and domain. After you have configured trust relationships between providers, changing these settings invalidates the trust model and requires a reimport of the provider's metadata.

   Modifying the base URL also invalidates the trust between the embedded service provider of the Access Gateway. To re-establish the trust after modifying the base URL, you have to restart the embedded service provider.

Base URL: The application path for the Identity Server. The Identity Server protocols (Liberty 1.2, SAML 1.1, and SAML 2.0) rely on this base URL to generate URL endpoints for each protocol.

- **Protocol:** The communication protocol. Specify HTTPS in order to run securely (in SSL mode) and for provisioning. Use HTTP only if you do not require security.
- **Domain:** The DNS name assigned to the Identity Server. When you are using an L4 switch, this DNS name should resolve to the virtual IP address set up on the L4 switch for the Identity Servers. Using an IP address is not recommended.
- **Port:** The port value for the protocol. Default ports are 8080 for HTTP or 8443 for HTTPS. If you want to use port 80 or 433, specify the port here, then configure the operating system to translate the port. See Section 38.4, "Translating the Identity Server Configuration Port," on page 596.
- **Application:** The Identity Server application. Leave the default value *nidp*.

SSL Certificate: Displays the Keystore page that you use to locate and replace the test-connector SSL certificate for this configuration.

The Identity Server comes with a test-connector certificate that you must replace for your production environment. You can replace the test certificate now or after you configure the Identity Server. If you create the certificate and replace the test-connector now, you can save some time by restarting Tomcat only once. Tomcat must be restarted whenever you assign an Identity Server to a configuration and whenever you update a certificate key store. See Section 6.5.3, "Managing the Keys, Certificates, and Trust Stores," on page 80.

LDAP Access: The maximum number of LDAP connections allowed to the configuration store. You can adjust this amount for system performance.

Session Timeout: The session inactivity time allowed before timing out. This is a global setting that applies to any resource that authenticates to this Identity Server or Identity Server cluster. The default setting is fifteen minutes.

This is a security setting:

- Lower it if you want idle sessions to time out with a smaller window of opportunity that allows someone to take over a session of a user who takes a break, leaving an active session unattended.
- Increase it if you want to allow idle users to have a longer time period before they are forced to log in again.

If the resource is configured to use Basic authentication or SSL mutual authentication, the session times out, but the browser must be closed to terminate the session.

Allow multiple browser session logout: Specifies whether a user with more than one session to the server is presented with an option to log out of all sessions. If you do not select this option, only the current session can be logged out. You deselect this option in instances where multiple users log in as guests. Then, when one user logs out, none of the other guests are logged out.

After you enable this option and click *OK*, you are prompted to apply the changes by using *Update Servers* on the Servers page. You must also restart any ESPs in an Access Gateway or J2EE Agent configuration that use this Identity Server configuration.

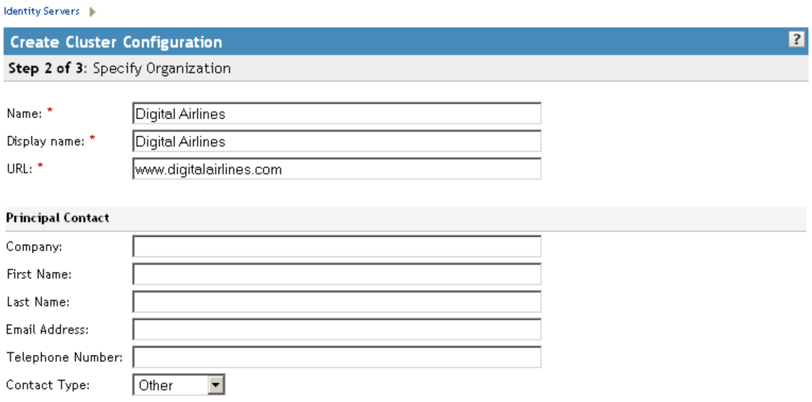
6. (Optional) If you are configuring the Identity Server for federation, either as an identity provider or an identity consumer, you might want to configure the following options. Otherwise, you can skip them.

   - "Configuring the General Identity Provider Options" on page 144

- “Configuring the General Identity Consumer Options” on page 145

7 To continue creating the Identity Server configuration, click *Next*.

The system displays the Organization page.



Step 2 of 3: Specify Organization

Name: * Digital Airlines
Display name: * Digital Airlines
URL: * www.digitalairlines.com

Principal Contact

Company:
First Name:
Last Name:
Email Address:
Telephone Number:
Contact Type: Other

Use this page to specify organization information for the Identity Server configuration. The information you specify on this page is published in the metadata for the Liberty 1.2 and SAML protocols. The metadata is traded with federation partners and supplies various information regarding contact and organization information located at the Identity Server.

The following fields require information:

- **Name:** The name of the organization.
- **Display Name:** The display name for the organization.
- **URL:** The organization’s URL for contact purposes.

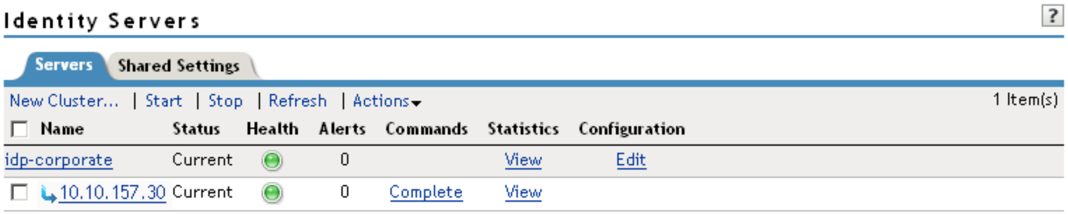
Optional fields include Company, First Name, Last Name, Email, Telephone, and Contact Type.

8 Click *Next* to configure the user store.

You must reference your own user store and auto-import the SSL certificate. See Section 8.1, “Configuring Identity User Stores,” on page 90 for information about this procedure.

9 After you configure the user store, click *Finish* to save the server configuration.

The system displays the new configuration on the Servers page.



☐ Name	Status	Health	Alerts	Commands	Statistics	Configuration
idp-corporate	Current	●	0		View	Edit
☐ 10.10.157.30	Current	●	0	Complete	View	

The status icons for the configuration and the Identity Server should turn green. It might take several seconds for the Identity Server to start and for the system to display a green light. If it does not, it is likely that the Identity Server is not communicating with the user store you set up. Ensure that you have entered the user store information correctly, and that you imported the SSL certificate to the user store. (*Edit > Local > [User Store]*.)

6.1.2 Assigning an Identity Server to a Cluster Configuration

After you create a configuration, you must assign the Identity Server to it. For clustering, you can assign more than one Identity Server to the configuration (see Section 6.1.4, "Managing a Cluster with Multiple Identity Servers," on page 60 for the steps to set up a cluster). A configuration uses any shared settings you have specified, such as attribute sets, user matching expressions, and custom attributes that are defined for the server.

1. In the Administration Console, click *Access Manager* > *Identity Servers*.
2. On the Servers page, select the server's check box, then choose *Actions* > *Assign to Cluster*.

   You can also select all displayed servers by selecting the top-level Server check box.
3. Select the configuration's check box, then click *Assign*.

   You are prompted to restart Tomcat. The status icon for the Identity Server should turn green. It might take several seconds for the identity provider to start and for the system to display the green light.

6.1.3 Removing a Server from a Configuration

Removing an Identity Server from a configuration disassociates the Identity Server from the cluster configuration. The configuration, however, remains intact and can be reassigned later or assigned to another server.

1. In the Administration Console, click *Access Manager* > *Identity Servers*.
2. Select the server, then click *Stop*. Wait for the Health indicator to turn red.
3. Select the server, then choose *Actions* > *Remove from Cluster*.

For information about deleting an Identity Server, see Section 5.1, "Managing an Identity Server," on page 49.

6.1.4 Managing a Cluster with Multiple Identity Servers

To add capacity and for system failover, you can cluster a group of Identity Servers and configure them in a cluster configuration to act as a single server. However, a cluster is not intended for login failover because all authentication data for a user is stored in memory on the cluster member or authenticating server that originally handled the user's authentication. If this server malfunctions, all users whose authentication data resides on the authenticating server must reauthenticate.

All requests that require user authentication information must be processed on the user's authenticating server. For example, if an HTTP request is received by a cluster server other than the authenticating server, then the HTTP request is forwarded to the authenticating server in the cluster. This server processes the HTTP request and routes it back through the forwarding cluster member and then to the original requester.

A cluster of Identity Servers should reside behind an L4 switch. Clients access the virtual IP (VIP) address of the cluster presented on the L4 switch, and the L4 switch alleviates server load by balancing traffic across the cluster. Whenever a user accesses the virtual IP address (port 8080) assigned to the L4, the system routes the user to one of the Identity Servers in the cluster, as traffic necessitates.

- "Prerequisites" on page 61

- “Setup” on page 61

Prerequisites

- ☐ An L4 switch installed. You can use the same switch for Identity Server clustering and Access Gateway clustering, provided that you use different virtual IPs. The LB algorithm can be anything (hash/sticky bit), defined at the Real server level. For configuration tips, see “Configuration Tips for the L4 Switch ” in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.
- ☐ Persistence (sticky) sessions enabled on the L4 server. Normally you define this at the virtual server level.
- ☐ An Identity Server configuration created for the cluster. You assign all the Identity Servers to this configuration. See Section 6.1.1, “Creating a Cluster Configuration,” on page 56 for information about creating an Identity Server configuration. See Section 6.1.2, “Assigning an Identity Server to a Cluster Configuration,” on page 60 for information about assigning identity servers to configurations.

  The base URL DNS name of this configuration must resolve via DNS to the IP address of the L4 virtual IP address. The L4 balances the load between the identity servers in the cluster.
- ☐ Ensure that the L4 administration server using port 8080 has the following TCP ports open:
  - 8443 (secure Administration Console)
  - 7801 range (for back-channel communication with cluster members. You need to open a port for each member of the cluster plus one. Thus, for a two member cluster, 7801, 7802, and 7803 need to be open.)
  - 636 (for secure LDAP)
  - 389 (for clear LDAP, loopback address)
  - 524 (network control protocol on the L4 machine for server communication)

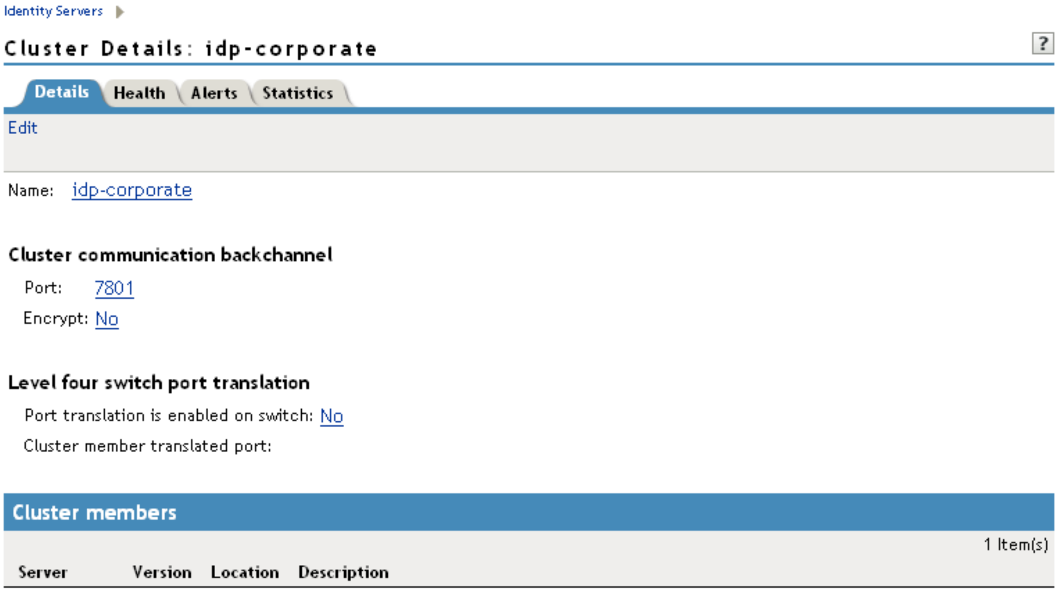
  The identity provider ports must also be open:
  - 8080 (non-secure login)
  - 8443 (secure login)
  - 1443 (server communication)

  If you are using introductions (see Section 6.1.1, “Creating a Cluster Configuration,” on page 56), you must configure the L4 switch to load balance on ports 8445 (identity provider) and 8446 (identity consumer).

Setup

1. Install the additional Identity Servers.

   During installation, choose option 2, *Install Novell Identity Server*. You run the installation for each new Identity Server you want to add. Specify the IP address and administration credentials of each additional Identity Server. If you are installing on a machine without the Administration Console, the installation asks you for the Administration Console’s IP address. After you install the Identity Servers, the servers are displayed on the Servers page in Identity Servers.
2. Assign the Identity Servers to the same cluster configuration (see Section 6.1.2, “Assigning an Identity Server to a Cluster Configuration,” on page 60).
3. Click the name of the cluster configuration.



Cluster Details: idp-corporate

Name: idp-corporate

Cluster communication backchannel

Port: 7801

Encrypt: No

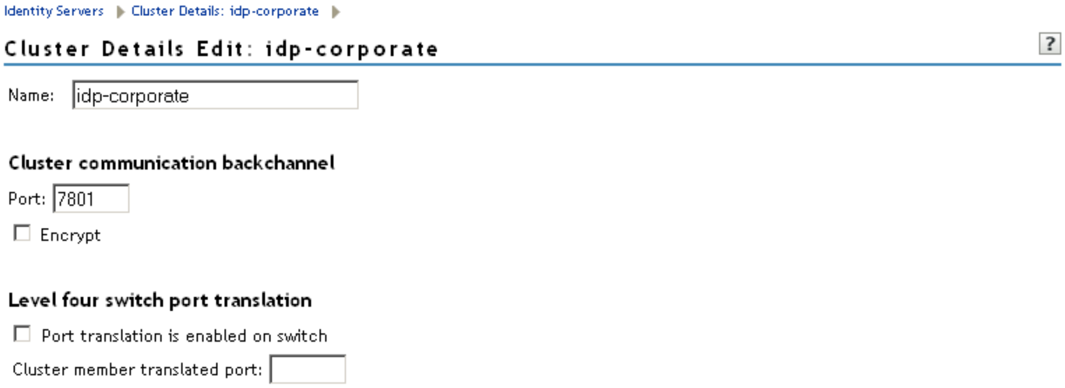
Level four switch port translation

Port translation is enabled on switch: No

Cluster member translated port:

The system displays the Cluster Details page, which lets you manage the configuration's cluster details, health, alerts, and statistics.

4 Click *Edit*.



Cluster Details Edit: idp-corporate

Name: idp-corporate

Cluster communication backchannel

Port: 7801

☐ Encrypt

Level four switch port translation

☐ Port translation is enabled on switch

Cluster member translated port:

5 Fill in the following fields as required:

Cluster Communication Backchannel: Provides a communications channel over which the cluster members maintain the integrity of the cluster. For example, this TCP channel is used to detect new cluster members as they join the cluster, and to detect members that leave the cluster. A small percentage of this TCP traffic is used to help cluster members determine which cluster member would best handle a given request. This back channel should not be confused with the IP address/port over which cluster members provide proxy requests to peer cluster members.

- **Port:** Specifies the TCP port of the cluster back channel on all of the Identity Servers in the cluster. 7801 is the default TCP port.

  Because the cluster back channel uses TCP, you can use cluster members on different networks. However, firewalls must allow the ports specified here to pass through. You need to open a port for each member of the cluster plus one. For example, if you use four devices, your port numbers would be 7801, 7802, 7803, 7804, and 7805.

  - **Encrypt:** Encrypts the content of the messages that are sent between cluster members.

  Level Four Switch Port Translation: Configures the L4 switch to translate the port of the incoming request to a new port when the request is sent to a cluster member. Because the cluster members communicate with each other over the same IP address/port as the L4 switch, the cluster implementation needs to know what that port is. The translated port is the port on the cluster members where other cluster members can contact it. This is the IP address and port where cluster members provide proxy requests to other cluster members.

  - **Port translation is enabled on switch:** Specifies whether the port of the L4 switch is different from the port of the cluster member. For example, enable this option when the L4 switch is using port 443 and the Identity Server is using port 8443.
  - **Cluster member translated port:** Specifies the port of the cluster member.

  Under *Cluster Members*, you can refresh, start, stop, and assign servers to Identity Server configurations.

6. Click *OK*, then update the Identity Server as prompted.

6.2 Modifying the Base URL

When configuring an Identity Server, you must carefully determine your settings for the base URL, protocol, and domain. Changing the base URL invalidates the trust model and requires a reimport of the provider's metadata, and a restart of the affected Access Gateway embedded service providers. It also changes the ID of the provider and the URLs that others use for access.

When you change the base URL of the Identity Server, you invalidate the following trusted relationships:

- The trusted relationships that the Identity Server has established with each Access Manager device that has been configured to use the Identity Server for authentication
- The trusted relationship that each Access Manager device has established with the Identity Server when the Identity Server configuration was selected.
- The trusted relationships that the Identity Server has established with other service providers.

The sessions of any logged in users are destroyed and no user can log in and access protected resources until the trust relationships are re-established.

To modify the base URL and re-establish trust relationships:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.
2. Change the protocol, domain, port, and application settings, as necessary.
3. Click *OK*.
4. On the Identity Servers page, click *Update*.

   This re-creates the trusted Identity Server configuration to use the new Base URL and metadata.
5. Restart Tomcat on each Identity Server in the configuration. Go to each machine, then enter the following command.

   ```
   /etc/init.d/novell-tomcat4 restart
   ```

6. For each Access Manager device configured to trust the configuration of this modified base URL, you must update the device so that the embedded service provider trusts the new Identity Server configuration:
   - Click *Access Gateways*, then click *Update* on any servers with a *Status* of *Update*.
   - Click *SSL VPNs*, then click *Update* on any servers with a *Status* of *Update*.
   - Click *J2EE Agents*, then click *Update* on any agents with a *Status* of *Update*.
7. For each service provider you have configured to trust the configuration of this modified base URL, you must send them the new metadata and have them re-import it.

For information about setting up SSL and changing an Identity Server from HTTP to HTTPS, see "Enabling SSL Communication" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

6.3 Enabling Role-Based Access Control

Role-based access control is used to provide a convenient way assign a user to a particular job function or set of permissions within an enterprise, in order to control access. In Access Manager, you assign users to roles, based on attributes of their identity, and then associate authorization policies to the role.

For a complete discussion on creating and configuring role policies, see Chapter 27, "Creating Role Policies," on page 391, in Part VI, "Policy Management," on page 383.

In order for a role to be assigned to users at authentication, you must enable it for the Identity Server configuration.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Roles*.
2. Click the role policy's check box, then click *Enable*.
3. To disable the role policy, click the role policy's check box, then click *Disable*.
4. After enabling or disabling role policies, update the Identity Server configuration on the Servers tab.

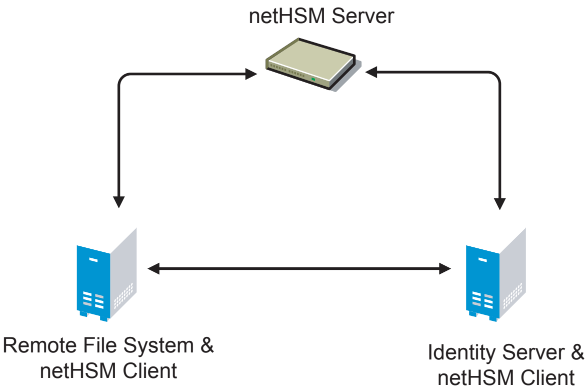
6.4 Using netHSM for the Signing Key Pair

The netHSM* is a Hardware Security Module (HSM) from nCipher*. The module is attached to the network and provides cryptographic resources for multiple servers. Keys stored in a netHSM keystore are secure because the key material can never be exposed outside of the module.

Access Manager has not been tested with any other HSM products; it has only been tested with the netHSM module from nCipher.

Figure 6-2 illustrates a simple netHSM configuration with an Identity Server as a netHSM client.

Figure 6-2 *A Simple netHSM Configuration*



Access Manager allows you to use netHSM to store and manage the signing key pair of the Identity Server. You must use the Administration Console to store and manage the other Access Manager certificates. Access Manager uses the Java Security provider of the netHSM server to interact with the netHSM server.

This section describes the following about the netHSM implementation:

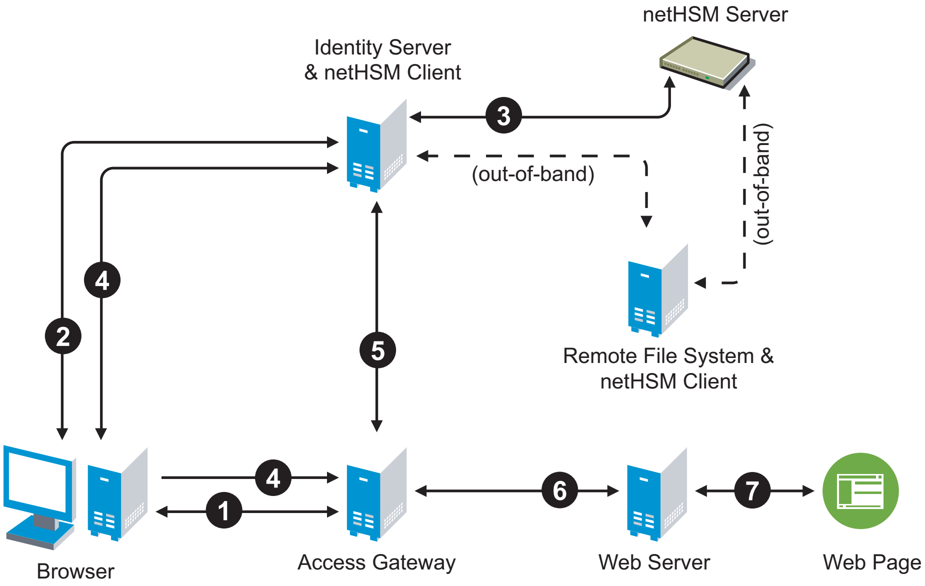
- Section 6.4.1, "Understanding How Access Manager Uses Signing and Interacts with the netHSM Server," on page 65
- Section 6.4.2, "Configuring the Identity Server for netHSM," on page 66

6.4.1 Understanding How Access Manager Uses Signing and Interacts with the netHSM Server

The netHSM server provides a signing certificate that is used instead of the one provided by Access Manager. Requests, responses, assertions, or payloads can be signed when there are interactions during single sign-on or during attribute queries between service providers and identity providers using any of the SAML1.1, SAML2, Liberty ID-FF, Liberty ID-WSF, or ID-SIS protocols. For more information about the services that use the signing certificate, see "Viewing the Services That Use the Signing Key Pair" on page 79.

Figure 6-3 outlines one of the basic flows that might occur during single sign-on to the Identity Server when authentication requests have been configured for signing.

Figure 6-3 Basic Flow for an Authentication Request Using netHSM



1. The user requests the Access Gateway to provide access to a protected resource.
2. The Access Gateway redirects the user to the Identity Server, which prompts the user for a username and password.
3. The Identity Server authenticates the user. If signing is enabled, the payload is signed by the netHSM server through the Java JSSE security provider.
4. The Identity Server returns the authentication artifact to the Access Gateway.
5. The embedded service provider of the Access Gateway retrieves the user's credentials from the Identity Server.
6. The Access Gateway verifies that the credentials allow the user access to the resource, then sends the request to the Web server.
7. The Web server returns the requested Web page.

6.4.2 Configuring the Identity Server for netHSM

The configuration tasks for netHSM are described in the following sections:

- "Prerequisites for Using netHSM" on page 66
- "Configuring the Identity Server to Be a netHSM Client" on page 67
- "Creating the nCipher Signing Key Pair" on page 68
- "Configuring the Identity Server to Use the netHSM Certificate" on page 73
- "Verifying the Use of the nCipher Key Pair" on page 75
- "Troubleshooting the netHSM Configuration" on page 75

Prerequisites for Using netHSM

☐ An installed and configured netHSM server.

☐ An installed and configured remote file system with the netHSM client.

☐ An installed Identity Server, assigned to a cluster configuration.

For instructions on a basic setup that assigns the Identity Server to a cluster configuration, see "Creating a Basic Identity Server Configuration" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

The following instructions describe one way to integrate the Identity Server with a netHSM server. Other ways are possible.

Configuring the Identity Server to Be a netHSM Client

The following instructions are based on nCipher hardware, but you should be able to adapt them for your hardware. The instructions explain how to configure the Identity Server so that it can communicate with both the nCipher server and the remote file system server, how to create a signing key pair and its keystore, how to copy these them to the Identity Server, and how to synchronize the changes with the remote file system server.

1 At the Identity Server, log in as `root` and install the netHSM client software.

The nCipher software installs files in the `/opt/nfast` directory and creates an nfast user and group. Check your netHSM documentation for the specific steps.

2 (Conditional) If your Identity Server cluster configuration contains more than one Identity Server, install the netHSM client software on the other Identity Servers in the cluster.

3 At the netHSM server, configure the server to allow the Identity Server to be a client.

Check your netHSM documentation for the specific steps.

4 (Conditional) If your Identity Server cluster configuration contains more than one Identity Server, configure the netHSM server to allow the other Identity Servers in the cluster to be a client.

5 At the Identity Server, enroll the client to use the server:

5a To get the ESN and hash numbers for the enroll command, enter the following command:

```
/opt/nfast/bin/anonkneti <IP_address>
```

Replace <*IP_address*> with the IP address of the netHSM server.

5b To enroll the client, enter the following command:

```
/opt/nfast/bin/nethsmenroll -p <IP_address> <ESN> <hash>
```

Replace <*IP_address*> with the IP address of the netHSM server. Replace <*ESN*> and <*hash*> with the values copied from the `anonkneti` command.

6 (Conditional) If the Identity Server and the Administration Console are installed on the same machine, modify the 9000 and 9001 TCP ports:

6a In a text editor, open the `/opt/novell/devman/share/conf/sc.conf` file.

6b Change the ports from 9000 and 9001 to another value, such as 9010 and 9011.

The lines should look similar to the following:

```
<stringParam name="ExecutorPort" value="9010" />
<stringParam name="SchedulerPort" value="9011" />
```

6c Save the changes.

6d Use the following command to restart Tomcat:

```
/etc/init.d/novell-tomcat4 restart
```

6e (Conditional) If other Identity Servers in the cluster contain an Administration Console, repeat Step 6.

7 At the Identity Server, enable the netHSM client so that it uses TCP:

7a Enter the following command:

```
/opt/nfast/bin/config-serverstartup -sp
```

7b To restart the nfast client, enter the following command:

```
/opt/nfast/sbin/init.d-nfast restart
```

8 Configure communication to the remote file system server. In this sample configuration the remote file system is installed on a Windows machine and the Identity Server is installed on Linux.

8a At the remote file system server, enable communication with the Identity Server. For a Windows machine, enter the following command:

```
C:\nfast\bin\rfs-setup.exe --gang-client --write-noauth
<address>
```

Replace <*address*> with the IP address of the Identity Server.

8b At the Identity Server, enable communication with the remote file system server. For nCipher, enter the following command:

```
/opt/nfast/bin/rfs-sync --setup --no-authenticate <address>
```

Replace <*address*> with the IP address of the remote file system server.

8c At the Identity Server, initialize synchronization with the remote file system server. For nCipher, enter the following commands:

```
/opt/nfast/bin/rfs-sync --update
/opt/nfast/bin/rfs-sync --commit
```

The first command reads updates from the remote file system server and downloads files to the `/opt/nfast/kmdata/local` directory. The second command writes local changes to the remote file system server.

9 Continue with "Creating the nCipher Signing Key Pair" on page 68.

Creating the nCipher Signing Key Pair

IMPORTANT: Because of Access Manager configuration conflicts, you need to use a netHSM client other than the Identity Server. The remote file system server is a netHSM client, or if you have configured another device as a client, you can use that device.

The following commands are specific to nCipher; it does not come with a tool to generate a key pair and CSR. nCipher also uses a unique keystore of type `nCipher.sworld`.

nCipher supports both a Windows and an Linux netHSM client.

- If you have a Windows netHSM client, the command is located in the following directory:
  `c:\Program Files\Java\jdk1.5.0_14\jre\bin\java`
- If you have Linux netHSM client, the command is located in the following directory:
  `/opt/novell/java/bin/java`

To create a new key pair for nCipher:

1 On a netHSM client, add the nCipher provider to the provider list of the `java.security` file:

  1a In a text editor, open the `C:\Program Files\Java\jdk1.5.0_14\jre\lib\security\java.security` file.

  1b Add the following lines to the top of the list of providers:

```
security.provider.1=com.ncipher.fixup.provider.nCipherRSAPrivat
eEncrypt
security.provider.2=com.ncipher.provider.km.nCipherKM
```

  The provider section should look similar to the following:

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=com.ncipher.fixup.provider.nCipherRSAPrivat
eEncrypt
security.provider.2=com.ncipher.provider.km.nCipherKM
security.provider.3=sun.security.provider.Sun
security.provider.4=sun.security.rsa.SunRsaSign
security.provider.5=com.sun.net.ssl.internal.ssl.Provider
security.provider.6=com.sun.crypto.provider.SunJCE
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
```

  1c Save your changes.

2 Add the nfast libraries to the CLASSPATH for Java:

For a Windows client, add the following paths:

```
c:\nfast\java\classes\keysafe.jar;c:\nfast\java\classes\nfjava.jar
;c:\nfast\java\classes\kmjava.jar;c:\nfast\java\classes\kmcsp.jar;
c:\nfast\java\classes\jutils.jar;c:\nfast\java\classes\jcetools.
jar;c:\nfast\java\classes\spp.jar;c:\nfast\java\classes\rsaprivenc
.jar;
```

For a Linux client, add the following paths and export them:

```
/opt/nfast/java/classes/nfjava.jar:/opt/nfast/java/classes/
kmjava.jar:/opt/nfast/java/classes/kmcsp.jar:/opt/nfast/java/
classes/spp.jar:/opt/nfast/java/classes/rsaprivenc.jar:/opt/nfast/
java/classes/jutils.jar:/opt/nfast/java/classes/jcetools.jar:/opt/
nfast/java/classes/keysafe.jar
```

3 Create a directory for the keystore and change to that directory.

4 On a Windows client, enter the following command to create a new key in a keystore:

```
"c:\Program Files\Java\jdk1.5.0_14\jre\bin\java" -Dprotect=module
-DignorePassphrase=true sun.security.tools.KeyTool -genkey -v
-alias od93 -keyalg RSA -keystore AMstore.jks -storetype
nCipher.sworld -provider com.ncipher.provider.km.nCipherKM
```

Enter your values for the following parameters:

Parameter	Description
`-Dprotect=module`	Only required if you want the keystore to be module protected.

Parameter	Description
`-DignorePassphrase=true`	Only required if you want the keystore to be module protected.
`sun.security.tools.KeyTool`	The name of the keytool command
`-alias`	A name that helps you identify the key. In this sample configuration, the name is `od93`.
`-keyalg`	The security algorithm.
`-keystore`	A name for the keystore. In this sample configuration, the name is `AMstore.jks`.
`-storetype`	The type of keystore. For nCipher, this must be set to `nCipher.sworld`.
`-provider`	The name of the providerClass and providerName. This is the provider that you added to the `java.security` file in Step 1.

The tool prompts you for a password for the keypass and the storepass. They must be the same password if you are going to use card set protection rather than module protection.

The tool also prompts you for the certificate subject name (first name, last name, organization, organizational unit, locality, state or providence, and country).

5 To generate a certificate request from a key in the keystore, enter the following command:

```
"c:\Program Files\Java\jdk1.5.0_14\jre\bin\java" -Dprotect=module
-DignorePassphrase=true sun.security.tools.KeyTool -certreq -alias
od93 -file cert.csr -keypass mypwd -keystore AMstore.jks -storepass
mypwd -storetype nCipher.sworld -provider
com.ncipher.provider.km.nCipherKM
```

Enter your values for the following parameters:

Parameter	Description
`-Dprotect=module`	Only required if you want the keystore to be module protected.
`-DignorePassphrase=true`	Only required if you want the keystore to be module protected.
`sun.security.tools.KeyTool`	The name of the keytool command
`-certreq`	The parameter that makes this a certificate request.
`-alias`	A name that helps you identify the certificate request. In this sample configuration, the name is od93.
`-file`	The name to be given to the certificate signing request file. In this sample configuration, the name is `cert.csr`.
`-keypass`	The password for the key. In this sample configuration, the password is mypwd.

Parameter	Description
`-keystore`	A name for the keystore. In this sample configuration, the name is `AMstore.jks`.
`-storepass`	The password for the keystore. In this sample configuration, the password is mypwd.
`-storetype`	The type of keystore. For nCipher, this must be set to `nCipher.sworld`.
`-provider`	The name of the providerClass and providerName.

6. Take the CSR created in Step 5 to a Certificate Authority. The CA needs to send you a der-encoded public certificate. The CA also needs to send you the public certificate that it used to create the certificate and the public certificates for any CAs in the chain.

7. Load the public certificate of the CA into the keystore by entering the following command:

```
"c:\Program Files\Java\jdk1.5.0_14\jre\bin\java" -Dprotect=module
-DignorePassphrase=true sun.security.tools.KeyTool -import -alias
publicca –file certca.cer -keystore Amstore.jks -storetype
nCipher.sworld -provider com.ncipher.provider.km.nCipherKM
```

Enter your values for the following parameters:

Parameter	Description
`-Dprotect=module`	Only required if you want the keystore to be module protected.
`-DignorePassphrase=true`	Only required if you want the keystore to be module protected.
`sun.security.tools.KeyTool`	The name of the keytool command
`-import`	The parameter that makes this an import request.
`-alias`	A name that helps you identify that this is the public certificate from the CA. In this sample configuration, the name is `publicca`.
`-file`	The name of the CA certificate file. In this sample configuration, the name is certca.cer.
`-keystore`	A name for the keystore. In this sample configuration, the name is `AMstore.jks`.
`-storetype`	The type of keystore. For nCipher, this must be set to `nCipher.sworld`.
`-provider`	The name of the providerClass and providerName.

The tool prompts you for the keystore password and asks whether you want to trust the certificate.

8. (Conditional) Repeat Step 7 for each CA in the chain, giving each CA a unique alias.

9. Import the signed certificated received from the CA by entering the following command:

   ```
   "c:\Program Files\Java\jdk1.5.0_14\jre\bin\java" -Dprotect=module
   -DignorePassphrase=true sun.security.tools.KeyTool -import -alias
   od93 -file signcert.der -keystore AMstore.jks -storepass mypwd
   -storetype nCipher.sworld -provider
   com.ncipher.provider.km.nCipherKM
   ```

   Enter your values for the following parameters:

Parameter	Description
`-Dprotect=module`	Only required if you want the keystore to be module protected.
`-DignorePassphrase=true`	Only required if you want the keystore to be module protected.
`sun.security.tools.KeyTool`	The name of the keytool command
`-import`	The parameter that makes this an import request.
`-alias`	A name that helps you identify that this is the signing key pair from the CA. It needs to be the same alias you specified when you created the keystore in Step 4. In this sample configuration, the name is `od93`.
`-file`	The name of the signing certificate file from the CA. In this sample configuration, the name is `signcert.der`.
`-keystore`	A name for the keystore. In this sample configuration, the name is `AMstore.jks`.
`-storepass`	The password for the keystore. In this sample configuration, the password is `mypwd`.
`-storetype`	The type of keystore. For nCipher, this must be set to `nCipher.sworld`.
`-provider`	The name of the providerClass and providerName.

10. (Optional) To verify that the certificates have been added to the keystore, enter the following command:

    ```
    "c:\Program Files\Java\jdk1.5.0_14\jre\bin\java" -Dprotect=module
    -DignorePassphrase=true sun.security.tools.KeyTool -list -v
    -keystore AMstore.jks -storetype nCipher.sworld -provider
    com.ncipher.provider.km.nCipherKM
    ```

    The keystore should contain at least two certificates. The certificate that you created should now be issued by the CA you used, and the public certificate of the CA should be there as the owner and the issuer.

11. Copy the keystore to the `/opt/novell/devman/jcc/certs/idp` directory on the Identity Server.

    The keystore is found on the netHSM client in the directory specified by the -keystore parameter when you created the keystore. See Step 4.

12 Synchronize the Identity Server with the remote file system server. For nCipher, enter the following commands:

```
/opt/nfast/bin/rfs-sync --update
/opt/nfast/bin/rfs-sync --commit
```

13 (Conditional) If the cluster configuration contains more than one Identity Server, complete the following steps for each cluster member:

13a Copy the keystore to the cluster member. Copy it to the following directory:

```
/opt/novell/devman/jcc/certs/idp
```

13b Make sure the `novlwww` user has at least read rights.

13c Use the netHSM client to synchronize the cluster member with the remote file system server. Enter the following commands:

```
/opt/nfast/bin/rfs-sync --update
/opt/nfast/bin/rfs-sync --commit
```

14 Continue with "Configuring the Identity Server to Use the netHSM Certificate" on page 73.

Configuring the Identity Server to Use the netHSM Certificate

1 At the Identity Server, log in as `root`.

2 Add the nfast jar files to the classpath.

Because the Identity Server runs as a Tomcat service, the following steps explain how to modify the classpath for Tomcat.

2a In an editor, open the `/opt/novell/tomcat4/bin/dtomcat4` file.

2b To the `CLASSPATH="$JAVA_HOME"/lib/tools.jar` line, add the following classes from the `/opt/nfast/java/classes` directory:

```
nfjava.jar
kmjava.jar
kmcsp.jar
spp.jar
rsaprivenc.jar
jutils.jar:
jcetools.jar
keysafe.jar
```

Your line should look similar to the following:

```
CLASSPATH="$JAVA_HOME"/lib/tools.jar:/opt/nfast/java/classes/
nfjava.jar:/opt/nfast/java/classes/kmjava.jar:/opt/nfast/java/
classes/kmcsp.jar:/opt/nfast/java/classes/spp.jar:/opt/nfast/
java/classes/rsaprivenc.jar:/opt/nfast/java/classes/
jutils.jar:/opt/nfast/java/classes/jcetools.jar:/opt/nfast/
java/classes/keysafe.jar
```

2c Save your changes.

3 Add the `novlwww` user to the `nfast` group by entering the following command:

```
usermod novlwww -G nfast
```

4 Add the netHSM certificate configuration lines to the `tomcat4.conf` file:

4a In a text editor, open the `/var/opt/novell/tomcat4/conf/tomcat4.conf` file.

4b Add the following lines:

```
JAVA_OPTS="${JAVA_OPTS} -Dcom.novell.nidp.extern.config.file=
/var/opt/novell/tomcat4/webapps/nidp/WEB-INF/classes/
externKeystore.properties"

JAVA_OPTS="${JAVA_OPTS} -Dprotect=module
-DignorePassphrase=true"
```

The first line specifies the location of the properties file. You can specify another location.

The second line is required only if you want the keystore to be module protected rather than card protected.

5 Configure the `externKeystore.properties` file to use the nCipher key and keystore:

5a In a text editor, create an `externKeystore.properties` file in the `/var/opt/novell/tomcat4/webapps/nidp/WEB-INF/classes` directory.

If you specified a different location for this file in Step 4, use that location.

5b Add the following lines:

```
com.novell.nidp.extern.signing.providerClass=com.ncipher.provid
er.km.nCipherKM
com.novell.nidp.extern.signing.providerName=nCipherKM
com.novell.nidp.extern.signing.keystoreType=nCipher.sworld
com.novell.nidp.extern.signing.keystoreName=/opt/novell/devman/
jcc/certs/idp/AMstore.jks
com.novell.nidp.extern.signing.keystorePwd=mypwd
com.novell.nidp.extern.signing.alias=od93
com.novell.nidp.extern.signing.keyPwd=mypwd
```

Enter your values for the following variables:

Variable	Value
`<provider_class>`	The name of the providerClass. For nCipher, this must be set to `com.ncipher.provider.km.nCipherKM`.
`<provider_name>`	The name of the provider. For nCipher, this must be set to `nCipherKM`.
`<keystore_type>`	The type of keystore. For nCipher, this must be set to `nCipher.sworld`.
`<keystore_name>`	The name you specified when you created the keystore. In this sample configuration, the name is `AMstore.jks`.
`<keystore_pwd>`	When using module-protected keys, the keystore password must be null. For example: `com.novell.nidp.extern.signing.keystorePwd=`
`<key_alias>`	The alias you created for the key when you created the key. In this sample configuration, the name is `od93`.
`<key_pwd>`	When using module-protected keys, the key password must be null. For example: `com.novell.nidp.extern.signing.keyPwd=`

6 To restart Tomcat, enter the following command:

```
/etc/init.d/novell-tomcat4 restart
```

7 Continue with "Verifying the Use of the nCipher Key Pair" on page 75.

Verifying the Use of the nCipher Key Pair

After you have configured the Identity Server to use the nCipher key pair and have restarted Tomcat, the metadata of the Identity Server indicates that the nCipher key pair is being used for the signing certificate.

1 In a browser, enter the following URL:

```
http://<DNS_name>:8080/nidp/idff/metadata
```

Replace <*DNS_name*> with the DNS name of your Identity Server.

2 Search for the following string:

```
<md:KeyDescriptor use="signing">
```

3 Copy the certificate text between the `<ds:X509Certificate>` and `</ds:X509Certificate>` tags

4 Paste the text into a text editor.

5 Delete the `<ds:X509Certificate>` tag and replace it with the following text:

```
-----BEGIN CERTIFICATE-----
```

6 Delete the `<ds:X509Certificate>` tag and replace it with the following text:

```
-----END CERTIFICATE-----
```

7 Save the file as a text file with a `.cer` extension.

8 Open the file in Internet Explorer.

9 View the certificate details.

If the Identity Server is using the nCipher signing certificate, the certificate is issued by your CA and the name the certificate is issued to is the name you specified for the certificate.

If the Identity Server is using the Access Manager certificate, the certificate is issued by the Organizational CA and the certificate name is test-signing. For troubleshooting information, see "Troubleshooting the netHSM Configuration" on page 75.

Troubleshooting the netHSM Configuration

To discover potential configuration errors:

1 Verify that you have not enabled the data encryption of resource IDs. There is a known issue with this feature and the Apache libraries in a multi-provider environment. Because of this issue, netHSM is not compatible with encrypting the resource IDs.

   1a In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Liberty* > *Web Service Provider*.

   1b Click a profile, then check the setting for the *Have Discovery Encrypt This Service's Resource Ids* option.

   1c If the option is selected, deselect it, then click *OK*.

   1d Verify that all profiles have been configured so that they do not encrypt the resource IDs.

2 View the nfast log files in the `/opt/nfast/log` directory.

When there is a port conflict, `logfile` contains entries similar to the following:

```
nFast server: Notice: Using tcp socket local:9000
nFast server: Fatal error during startup: Operating system call
failed: bind tcp socket, Address already in use
```

For information on how to change the port, see Step 6 on page 67. For other errors, consult the netHSM documentation.

3 If the `novlwww` user does not have rights to the `cmdadp.log` and `cmdadp-debug.log` files, the Identity Server is halted because it cannot read the keystore. The Health page of the Identity Server displays the following error:

```
The following error occurred during the identity server
configuration. Unable to read keystore: /opt/novell/devman/jcc/
certs/idp/AMstore45.jks
```

To correct the error:

3a View the rights for the nfast log files with the following command:

```
ll /opt/nfast/log
```

Your listing should look similar to the following:

```
-rw-r--r-- 1 novlwww nfast    0 Apr 11 11:50 cmdadp-debug.log
-rw-r--r-- 1 novlwww nfast  134 Apr 11 11:50 cmdadp.log
-rw-r----- 1 root    nfast   43 Apr 11 11:49 debug
-rw-r----- 1 nfast   nfast    5 Apr 11 11:49 hardserver.pid
-rw-r----- 1 nfast   nfast 3057 Apr 11 11:50 logfile
```

If `novlwww` is not listed as the owner of the `cmdadp.log` and `cmdadp-debug.log` files, continue with Step 3b.

If `novlwww` is listed as the owner of the files with rw permissions, log file ownership is not the source of your problem. Continue with Step 4.

3b Stop Tomcat with the following command:

```
/etc/init.d/novell-tomcat4 stop
```

3c Stop nfast with the following command:

```
/opt/nfast/sbin/init.d-nfast stop
```

3d Delete all the log files in the `/opt/nfast/log` directory.

3e Start nfast with the following command:

```
/opt/nfast/sbin/init.d-nfast start
```

3f Start Tomcat with the following command

```
/etc/init.d/novell-tomcat4 start
```

3g Wait a minute, then list the files in the `/opt/nfast/log` directory.

The nfast client creates the log files and assigns the correct owners and rights.

4 Enable Identity Server logging and view the `catalina.out` file.

4a In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Logging*.

4b Configure the following options:

File Logging: Specify enabled.

Echo to Console: Select this option.

Component File Logger Levels: Set *Application* to *debug*.

Trace Logging: Specify enabled.

Component Content Filters: Select *Application*.

4c Click *OK*, then update the Identity Server.

4d Delete the current `catalina.out` file in the `/var/opt/novell/tomcat4/logs` directory.

4e Restart Tomcat by entering the following command:

```
/etc/init.d/novell-tomcat4 restart
```

4f To tail the `catalina.out` file, enter the following command:

```
tail -f /var/opt/novell/tomcat4/logs/catalina.out
```

4g Search for a list of providers. When nCipher is working, the file contains entries similar to the following and nCipher entries:

```
Security Providers:
     SUN: 1.42
        SUN (DSA key/parameter generation; DSA signing; SHA-1,
MD5 digests; SecureRandom; X.509 certificates; JKS keystore;
PKIX CertPathValidator; PKIX CertPathBuilder; LDAP, Collection
CertStores)
     SunJSSE: 1.42
        Sun JSSE provider(implements RSA Signatures, PKCS12,
SunX509 key/trust factories, SSLv3, TLSv1)
     SunRsaSign: 1.42
        SUN's provider for RSA signatures
     SunJCE: 1.42
        SunJCE Provider (implements DES, Triple DES, AES,
Blowfish, PBE, Diffie-Hellman, HMAC-MD5, HMAC-SHA1)
     SunJGSS: 1.0
        Sun (Kerberos v5)
     nCipherRSAPrivateEncrypt: 1.008004
        RSA private key encrypt handling provider
     nCipherKM: 1.008004
        nCipher Secure Key Management
     BC: 1.28
        BouncyCastle Security Provider v1.28
     SAML: 1.0
        SAML SASL Mechanism
```

4h (Conditional) If the `catalina.out` file does not contain any entries for providers, check for the following errors:

- Check the Health of the Identity Server. If the status is red, use the error message to resolve the issue.
- Make sure the `novlwww` user has read rights to the keystore.
- Verify that the `externKeystore.properties` file has all the required lines with valid values. See Step 5 on page 74.
- Verify that the `tomcat4.conf` file is configured correctly. See Step 4 on page 73.

5 Enable netHSM logging.

This logging feature is very verbose. It should be turned on only while you are debugging a problem. If it is left on, your machine can quickly run out of disk space.

5a To the `tomcat4.conf` file in the `/var/opt/novell/tomcat4/conf` directory, add the following line:

```
JAVA_OPTS="${JAVA_OPTS} -DJCECSP_DEBUG=255 -DJCECSP_DEBUGFILE=/
var/opt/novell/tomcat4/logs/nCipher_jcecsp.debug"
```

5b Restart Tomcat by entering the following command:

```
/etc/init.d/novell-tomcat4 restart
```

5c Look for clues in the `nCipher_jcecsp.debug` file.

6.5 Configuring Secure Communication on the Identity Server

The Identity Server uses the following key pairs for secure communication. In a production environment, you should exchange the key pairs that are created at installation time with certificates from a trusted Certificate Authority.

- **Connector:** The test-connector key pair is used when you establish SSL communication between the Identity Server and the browsers and between the Identity Server and the Access Gateway back-channel communications. It needs to be replaced with a certificate that has a subject name that matches the DNS name of the Identity Server. This task is part of basic setup. See "Enabling SSL Communication" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.
- **Signing:** The test-signing key pair is used by the various protocols to sign authentication requests, to sign communication with providers on the SOAP back-channel, and to sign Web Service Provider profiles. For more information on the services that use the signing certificate, see Section 6.5.1, "Viewing the Services That Use the Signing Key Pair," on page 79.

  This certificate can be stored in an external HSM keystore. For information on how to use netHSM to replace and manage this signing certificate, see Section 6.4, "Using netHSM for the Signing Key Pair," on page 64.
- **Data Encryption:** The test-encryption key pair is used to encrypt specific fields or data in the assertions. For more information on the services that use the encryption certificate, see Section 6.5.2, "Viewing Services That Use the Encryption Key Pair," on page 80.

If you are going to use introductions in your federation configuration, you need to set up the following key pairs:

- **Identity provider:** The test-provider key pair is used when you configure your Identity Server to use introductions with other identity providers and have set up a common domain name for this purpose. It needs to be replaced with a certificate that has a subject name that matches the DNS name of the common domain. For configuration information, see Section 9.4.1, "Configuring the General Identity Provider Options," on page 144.
- **Identity consumer:** The test-consumer key pair is used when you configure your Identity Server to use introductions with other service providers and have set up a common domain name for this purpose. It needs to be replaced with a certificate that has a subject name that matches the DNS name of the common domain. For configuration information, see Section 9.4.2, "Configuring the General Identity Consumer Options," on page 145.

To enable secure communication between the user store and the Identity Server, you can also import the trusted root certificate of the user store. For configuration information, see Section 8.1.2, "Configuring the User Store," on page 91.

This section describes the following tasks:

- Section 6.5.1, "Viewing the Services That Use the Signing Key Pair," on page 79

- Section 6.5.2, “Viewing Services That Use the Encryption Key Pair,” on page 80
- Section 6.5.3, “Managing the Keys, Certificates, and Trust Stores,” on page 80

6.5.1 Viewing the Services That Use the Signing Key Pair

The following services can be configured to use signing:

- “Protocols” on page 79
- “SOAP Back Channel” on page 79
- “Profiles” on page 79

Protocols

The protocols can be configured to sign authentication requests and responses.

To view your current configuration:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.
2. In the *Identity Provider* section, view the setting for the *Require Signed Authentication Requests* option. If it is selected, all authentication requests from identity providers are signed.
3. In the *Identity Consumer* section, view the settings for the *Require Signed Assertion*s and S*ign Authentication Requests* options. If these options are selected, assertions and authentication requests are signed.

SOAP Back Channel

The SOAP back channel is the channel that the protocols use to communicate directly with a provider. The SOAP back channel is used for artifact resolutions and attribute queries for the Identity Web Services Framework.

To view your current configuration for the SOAP back channel:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.
2. Select the protocol (Liberty, SAML 1.1, or SAML 2.0), then click the name of an identity provider or service provider.
3. Click *Access*.
4. View the *Security* section. If the *Message Signing* option is selected, signing is enabled for the SOAP back channel.

Profiles

Any of the Web Service Provider profiles can be enabled for signing by configuring them to use X.509 for their message-level security mechanism.

To view your current configuration:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Liberty* > *Web Service Provider*.
2. Click the name of a profile, then click *Descriptions*.
3. Click the *Description Name*.

4. If either *Peer entity = None, Message=X509* or *Peer entity = MutualTLS, Message=X509* has been selected as the security mechanism, signing has been enabled for the profile.

6.5.2 Viewing Services That Use the Encryption Key Pair

All of the Liberty Web Service Provider Profiles allow you to configure them so that the resource IDs are encrypted. By default, no profile encrypts the IDs.

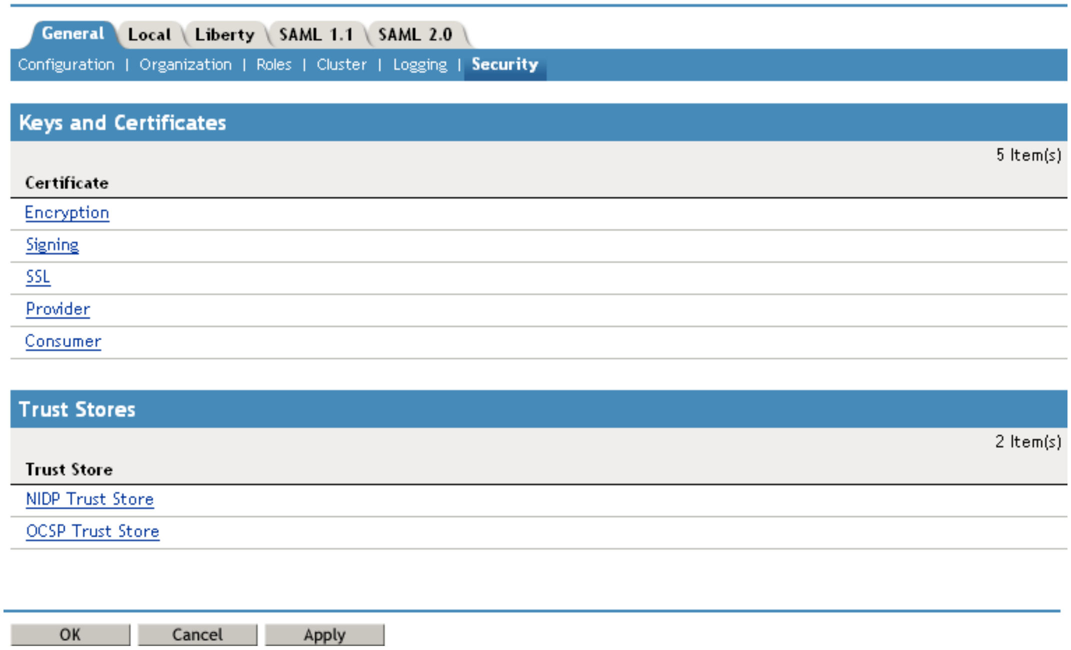
To view your current configuration:

1. In the Administration Console, click *Access Manager > Identity Servers > Edit > Liberty > Web Service Provider*.
2. Click the name of a profile.
3. If the *Have Discovery Encrypt This Service's Resource IDs* option is selected, the encryption key pair is used to encrypt the resource IDs.

6.5.3 Managing the Keys, Certificates, and Trust Stores

You can view the private keys, CA certificates, and certificate containers associated with the Identity Server configuration. Primarily, you use the Security page to add and replace CA certificates as necessary and to perform certificate management tasks, such as adding trusted root certificates to a trust store.

1. In the Administration Console, click *Access Manager > Identity Servers > Edit > Security*.


General
Local
Liberty
SAML 1.1
SAML 2.0
Configuration | Organization | Roles | Cluster | Logging | Security

Keys and Certificates

5 Item(s)

Certificate
Encryption
Signing
SSL
Provider
Consumer

Trust Stores

2 Item(s)

Trust Store
NIDP Trust Store
OCSP Trust Store

2. To view or manage keys and certificates:
   - **2a** Click any of the following links:

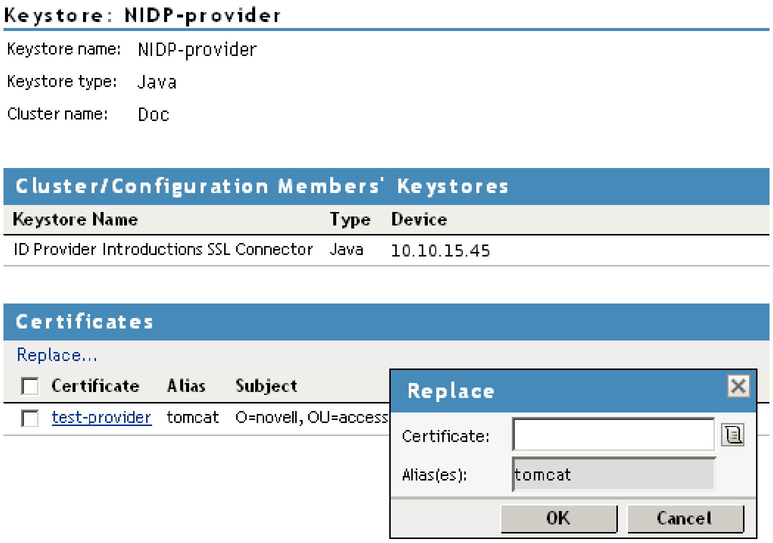
     Encryption: Displays the NIDP-encryption certificate keystore. The encryption certificate is used to encrypt specific fields or data in the assertions. Click *Replace* to replace the encryption certificate.

Signing: Displays the NIDP-signing certificate keystore. The signing certificate is used to sign the assertion or specific parts of the assertion. Click *Replace* to replace the signing certificate.

SSL: (Required) Displays the NIDP-connector keystore. Click this link to access the keystore and replace the connector certificate.

Provider: Displays the NIDP-provider keystore. Click this link to access the keystore and replace the provider certificate used by the Identity Server when it is acting as an identity provider.

Consumer: Displays the NIDP-consumer keystore. Click this link to access the keystore and replace the consumer certificate used by the Identity Server when it is acting as an identity consumer (service provider).



Keystore: NIDP-provider

Keystore name:	NIDP-provider
Keystore type:	Java
Cluster name:	Doc

Cluster/Configuration Members' Keystores

Keystore Name	Type	Device
ID Provider Introductions SSL Connector	Java	10.10.15.45

Certificates

☐ Certificate	Alias	Subject
☐ test-provider	tomcat	O=novell, OU=access

Replace

Certificate:

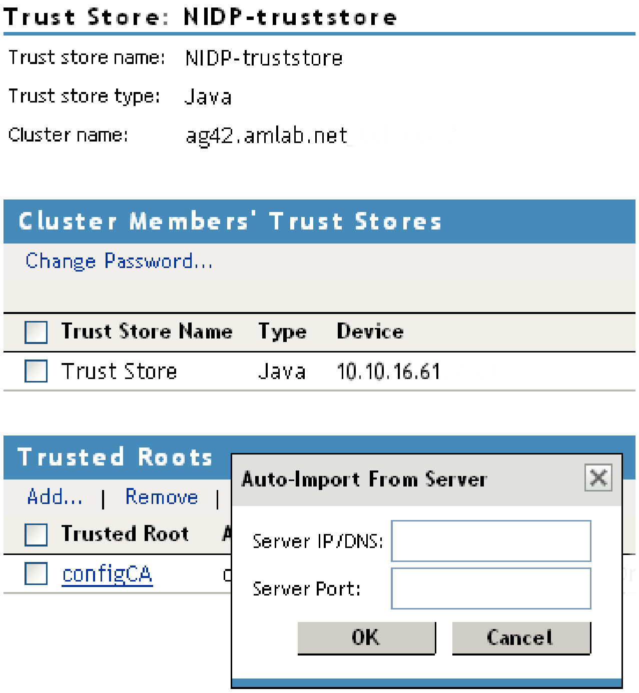
Alias(es): tomcat

2b To replace a certificate, click *Replace*, browse to locate the certificate, then click *OK*.

3 To manage trust stores associated with the Identity Server, click either of the following links on the Security page:

NIDP Trust Store: The trusted root certificate container for CA certificates associated with the Identity Server. Click this link to access the trust store, where you can change the password or add trusted roots to the container. Liberty and SAML 2.0 protocol messages that are exchanged between identity and service providers often need to be digitally signed. A provider uses the signing certificate included with the metadata of a trusted provider to validate signed messages from the trusted provider. To use SSL for protocol messages to be exchanged between providers, each provider must trust the SSL Certificate Authority (CA) of the other provider. Well-known CAs should already be trustable, but for those that are not, you must import the CA for the other provider. Failure to do so causes numerous system errors.

OCSP Trust Store: The trust store for OCSP certificates. Online Certificate Status Protocol is a method used for checking the revocation status of a certificate. To use this feature, you must set up an OCSP server. The Identity Server sends an OCSP request to the OCSP server to determine if a certain certificate has been revoked. The OCSP server replies with the revocation status. If this revocation checking protocol is used, the Identity Server does not cache or store the information in the reply, but sends a request every time it needs to check the revocation status of a certificate. The OCSP reply is signed by the OCSP server. To verify that it was signed by the correct OCSP server, the OCSP server certificate needs to be added to this trust store. The OCSP server certificate itself is added to the trust store, not the CA certificate.



Trust store name: NIDP-truststore
Trust store type: Java
Cluster name: ag42.amlab.net

☐ Trust Store Name	Type	Device
☐ Trust Store	Java	10.10.16.61

4 Specify the server IP address and port.

The auto-import displays the certificate chain, which you can select for import.

5 Click *OK*, then click *Close*.

6 Restart Tomcat.

The system prompts you with a dialog box to restart Tomcat. This is necessary whenever security changes are made to the Identity Server.

For more information about enabling security for a basic Access Manager configuration, see "Enabling SSL Communication" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

For additional information about managing certificates, see Part V, "Security and Certificate Management," on page 353.

Defining Shared Settings

7

You can define shared settings so that they can be reused and are available in any Identity Server cluster configuration. The settings include:

- **Attribute sets:** Sets of attributes that are exchangeable between identity and service providers.
- **User matching expressions:** The logic of the query to the user store for identification when an assertion is received from an identity provider.
- **SharedSecret names:** Custom shared secret names that you want to be available when configuring policies.
- **LDAP attributes:** Custom LDAP attribute names that you want to be available when configuring policies.

This section describes the settings that can apply to any configuration.

- Section 7.1, "Configuring Attribute Sets," on page 83
- Section 7.2, "Editing Attribute Sets," on page 85
- Section 7.3, "Configuring User Matching Expressions," on page 85
- Section 7.4, "Adding Custom Attributes," on page 86

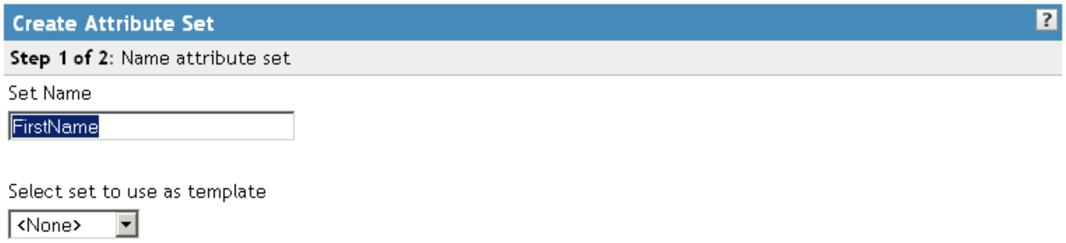
7.1 Configuring Attribute Sets

Attributes you specify on the Identity Server are used in attribute requests and responses, depending on whether you are configuring a service provider (request) or identity provider (response). Attribute sets provide a common naming scheme used in the exchange. For example, an attribute set can map the Liberty attribute FN (first name) to the equivalent remote name used at the service provider, which might be Name.

Attributes also can be defined and used in policy enforcement. They can be attributes defined by the Web Service Profiles, or customized attributes that can be mapped into SAML attributes. You also map user attributes so that the Identity Server can accept them from SAML.

To create and configure an attribute set:

1 In the Administration Console, click *Access Manager* > *Identity Server* > *Shared Settings* > *Attribute Sets* > *New*.



Create Attribute Set

Step 1 of 2: Name attribute set

Set Name

FirstName

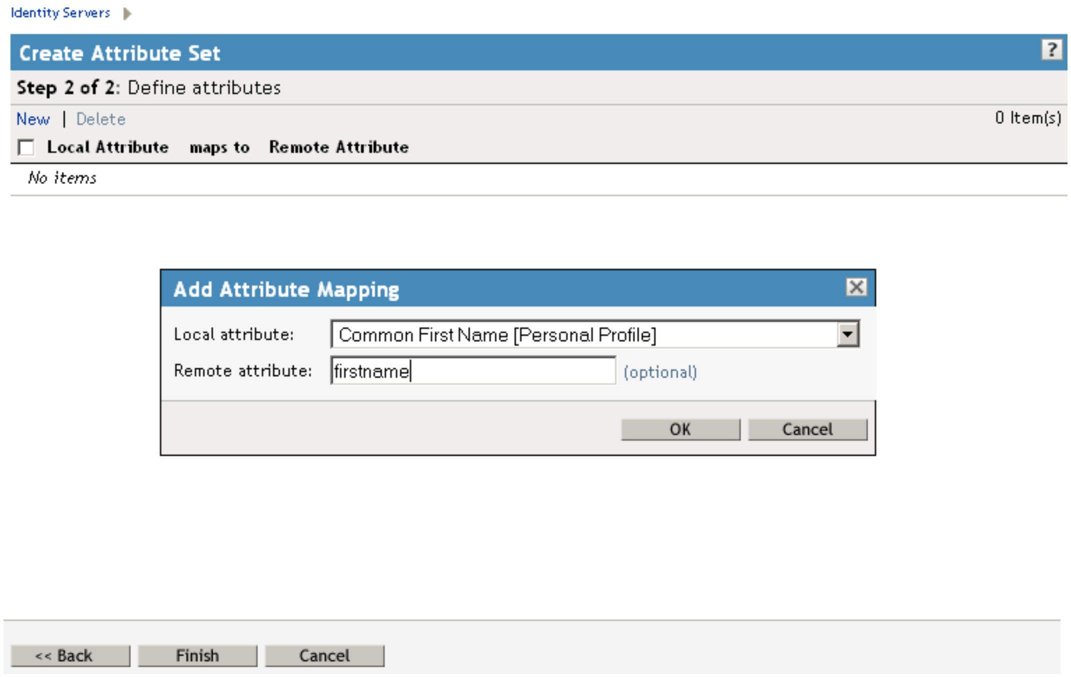
Select set to use as template

<None>

2 Specify a name for identifying the attribute set, then click *Next*.

You can select an existing attribute set that you have created, which you can use as a template for the new set.

3 To create a set, click *New*.



Local attribute: Common First Name [Personal Profile]

Remote attribute: firstname (optional)

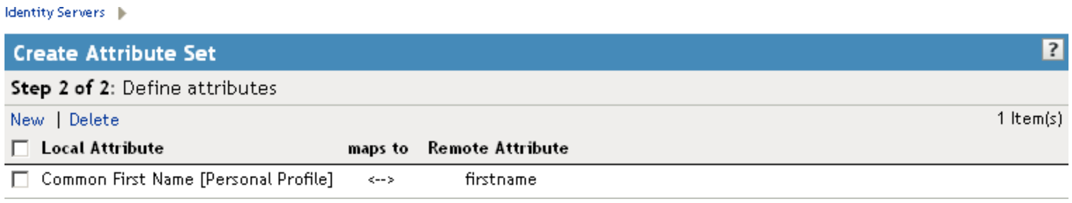
Local Attribute: A drop-down list of all server profile and LDAP attributes. As an example, you can select *All Roles* to use in role policies, which enables trusted providers to send role information in authentication assertions. Customizable attributes can be created and displayed in this list.

Remote Attribute: The name of the attribute defined at the external provider. The text for this field is case sensitive. If you leave this field blank, the system sends an internal value that is recognized between Identity Servers.

For a SAML 1.1 identity consumer (service provider), a name identifier received in an assertion is automatically given a remote attribute name of *saml:NameIdentifier.* This allows the name identifier to be mapped to a profile attribute that can then be used in policy definitions.

4 Click *OK*.

The system displays the map settings on the Define Attributes page, as shown below:

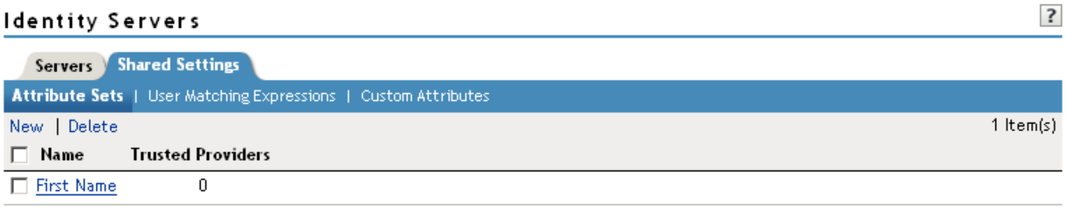


☐	Local Attribute	maps to	Remote Attribute
☐	Common First Name [Personal Profile]	<-->	firstname

You can continue adding as many attributes as you need.

5 Click *Finish* after you created the map.

The system displays the map on the Attribute Sets page, as well as indicating whether it is in use by a provider. (See Section 9.8, “Selecting Attributes for a Trusted Provider,” on page 155.)

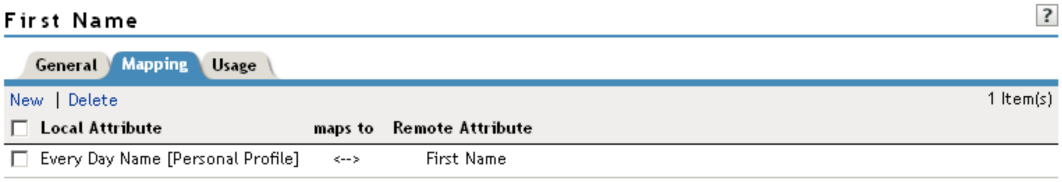


☐	Name	Trusted Providers
☐	First Name	0

7.2 Editing Attribute Sets

You can edit attribute sets that have been created in the system. (See Section 7.1, “Configuring Attribute Sets,” on page 83.)

1. In the Administration Console, click *Access Manager* > *Identity Server* > *Shared Settings* > *Attribute Sets*.
2. Click the name of the attribute set that you want to edit.



☐	Local Attribute	maps to	Remote Attribute
☐	Every Day Name [Personal Profile]	<-->	First Name

3. The system displays an attribute set page with the following tabs:

   General: Click to edit the name of the attribute set.

   Mapping: Click to edit the attribute map.

   Usage: Displays where the attribute set is used. Informational only.
4. Click *OK*, then click *Close*.

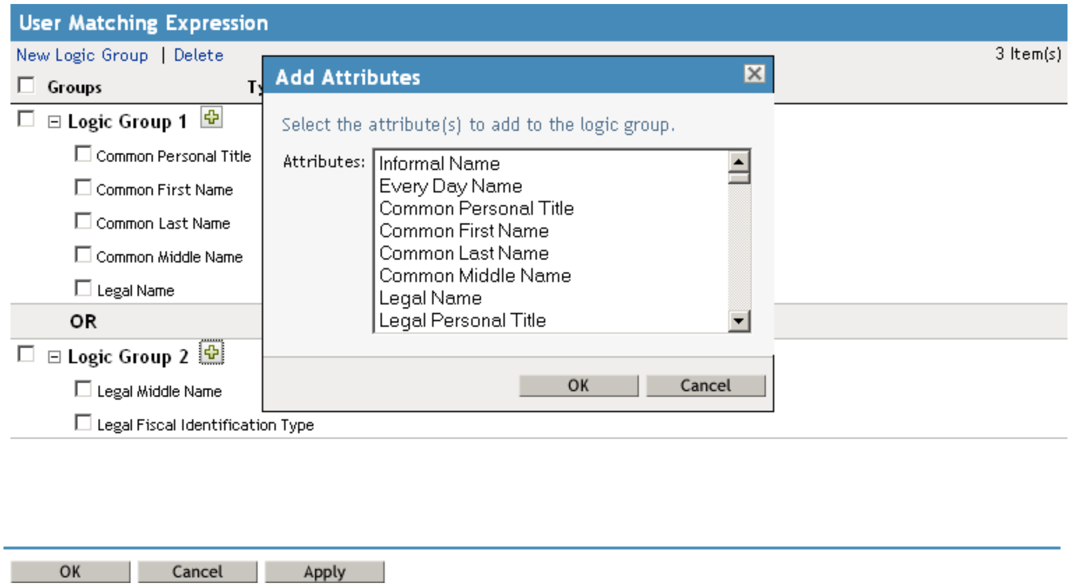
7.3 Configuring User Matching Expressions

One of the user identification methods the Identity Server uses when an assertion is received is to query the user store based on attributes received in the assertion from the identity provider. You configure user matching expressions to define the logic of the query. You must know the LDAP attributes that are used to name the users in the user store and create the user’s distinguished name.

In order to use user matching, you must enable the Personal Profile on the identity provider and the service provider. See Section 12.2, “Enabling Web Services and Profiles,” on page 174.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Shared Settings* > *User Matching Expressions*.
2. Click *New*, or click the name of an existing user matching expression.

   Name: The name of the user lookup expression.
3. Click the *Add Attributes* icon (plus sign), then select attributes to add to the logic group. (Use the Shift key to select several attributes.)



4 Click *OK*.

5 To add logic groups, click *New Logic Group*.

The *Type* drop-down (AND or OR) applies only between groups. Attributes within a group are always the opposite of the type selection. For example, if the *Type* value is AND, the attributes within the group are OR.

6 Click the *Add Attributes* icon (plus sign) to add attributes to the next logic group, then click *OK*.

7 Click *Finish*.

7.4 Adding Custom Attributes

You can add custom shared secret names or LDAP attribute names that you want to make available for selection when setting up policies.

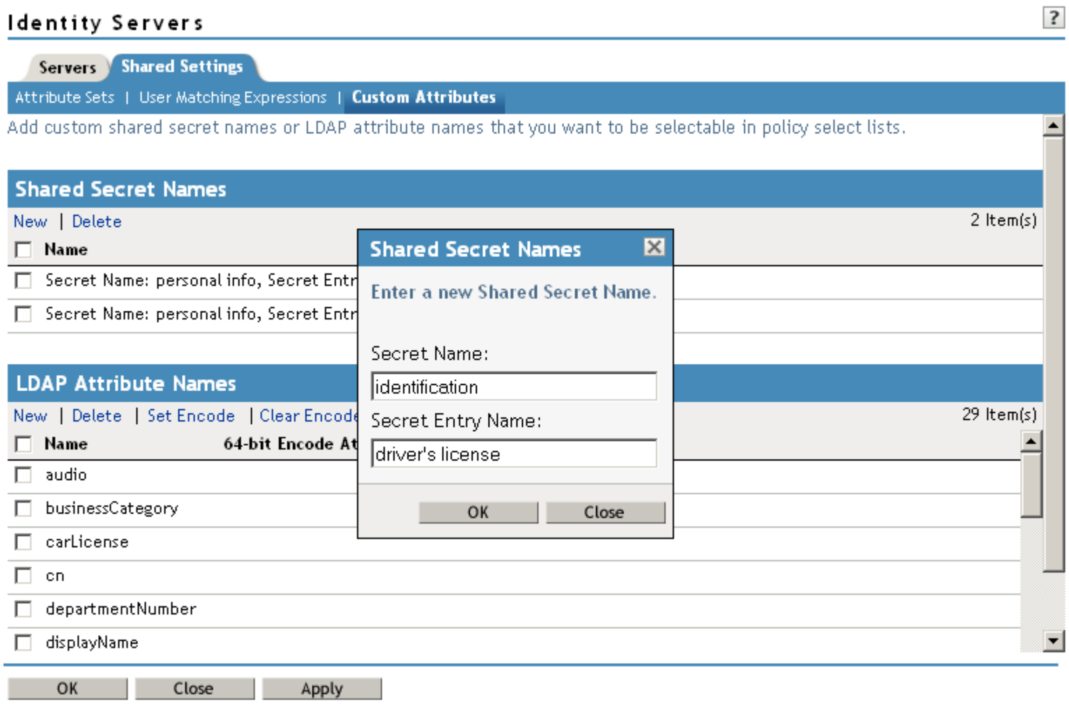
- Section 7.4.1, "Creating Shared Secret Names," on page 86
- Section 7.4.2, "Creating LDAP Attribute Names," on page 87

7.4.1 Creating Shared Secret Names

The shared secret consists of a secret name and one or more secret entry names. You can create a secret name only, or a secret name and an entry name. Shared secret names can be created either on this page or in the associated policy that consumes them.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Shared Settings* > *Custom Attributes*.

2 To create shared secret names, click *New*.



Add custom shared secret names or LDAP attribute names that you want to be selectable in policy select lists.

☐ Secret Name: personal info, Secret Entr

☐ Secret Name: personal info, Secret Entr

☐ audio

☐ businessCategory

☐ carLicense

☐ cn

☐ departmentNumber

☐ displayName

Shared Secret Names

Enter a new Shared Secret Name.

Secret Name: identification

Secret Entry Name: driver's license

3 Enter a new shared secret name and, optionally, a secret entry name.

4 Click *OK*.

7.4.2 Creating LDAP Attribute Names

LDAP attributes are available for all policies. You can add available attributes here, as well as on the Policies page. LDAP attribute names can be created either on this page or in the associated policy that consumes them.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Shared Settings* > *Custom Attributes*.

2 Click *New* to add a name. This list is customizable. Examples of predefined LDAP attributes include:

audio: Uses a u-law encoded sound file, stored in the directory.

businessCategory: Describes the kind of business performed by an organization.

carLicense: Vehicle license or registration plate.

cn: The X.500 commonName attribute, which contains a name of an object. If the object corresponds to a person, it is typically the person’s full name.

departmentNumber: Identifies a department within an organization.

displayName: The preferred name of a person to be used when displaying entries. Identifies a name to be used. When displaying an entry, especially within a one-line summary list, it is useful to use this value. Because other attribute types such as cn are multivalued, an additional attribute type is needed.

employeeNumber: Numerically identifies a person within an organization.

employeeType: Identifies the type of employee.

givenName: Identifies the person's name that is not his or her surname or middle name.

homePhone: Identifies a person by home phone.

homePostalAddress: Identifies a person by home address.

initials: Identifies a person by his or her initials. This attribute contains the initials of an individual, but not the surname.

jpegPhoto: Stores one or more images of a person, in JPEG format.

labeledURI: Uniform Resource Identifier with an optional label. The label describes the resource to which the URI points.

mail: A user's e-mail address.

manager: Identifies a person as a manager.

mobile: Specifies a mobile telephone number associated with a person.

o: The name of an organization.

pager: The pager telephone number for an object.

photo: Specifies a photograph for an object.

preferredLanguage: Indicates an individual's preferred written or spoken language.

roomNumber: The room number of an object.

secretary: Specifies the secretary of a person.

sn: The X.500 surname attribute, which contains the family name of a person.

uid: User ID.

userCertificate: An attribute stored and requested in the binary form.

userPKCS12: A format to exchange personal identity information. Use this attribute when information is stored in a directory service.

userSMIMECertificate: PKCS#7 SignedData used to support S/MIME. This value indicates that the content that is signed is ignored by consumers of userSMIMECertificate values.

x500uniqueIdentifier: Distinguishes between objects when a distinguished name has been reused. This is a different attribute type from both the *uid* and the *uniqueIdentifier* type.

3. To configure 64-bit attribute data encoding, click an attribute's check box, then click one of the following links:

   Set Encode: Specifies that LDAP returns a raw format of the attribute rather than binary format, which Access Manager encodes to base 64, so that the protected resource understands the attribute. You might use base 64 encoding if you use certificates that require raw bites rather than binary string format.

   Clear Encode: Deletes the 64-bit data encoding setting.

4. Click *Apply* to save changes, then click the *Servers* tab to return to the Servers page.

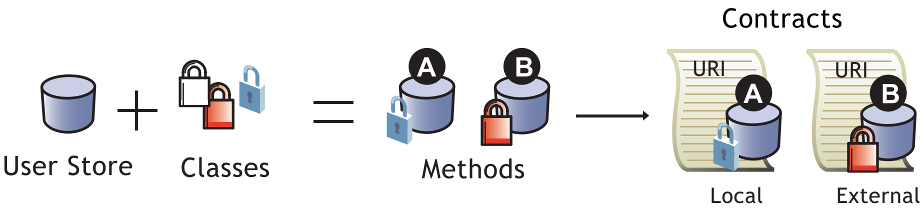
Configuring Local Authentication

8

To guard against unauthorized access, Access Manager supports a number of ways for users to authenticate. These include name/password, RADIUS token-based authentication, and X.509 digital certificates. You configure authentication at the Identity Server by creating authentication contracts that the components of Access Manager (such as an Access Gateway) can use to protect a resource.

Figure 8-1 illustrates the components of a contract:

Figure 8-1 *Local Authentication*



- **User stores:** The user directories to which users authenticate on the back end. You set up your user store when creating the Identity Server cluster configuration.
- **Classes:** The code (a Java class) that implements a particular authentication type (name/password, RADIUS, and X.509) or means of obtaining credentials. Classes specify how the Identity Server requests authentication information, and what it should do to validate those credentials.
- **Methods:** The pairing of an authentication class with one or more user stores, and whether the method identifies a user.
- **Contracts:** The basic unit of authentication. Contracts can be local (executed at the server) or external (satisfied by another Identity Server). Contracts are identified by a unique URI that can be used by Access Gateways and agents to protect resources. Contracts are comprised of one or more authentication methods used to uniquely identify a user. You can associate multiple methods with one contract.

You can also use the properties of a class to create custom login pages.

8.1 Configuring Identity User Stores

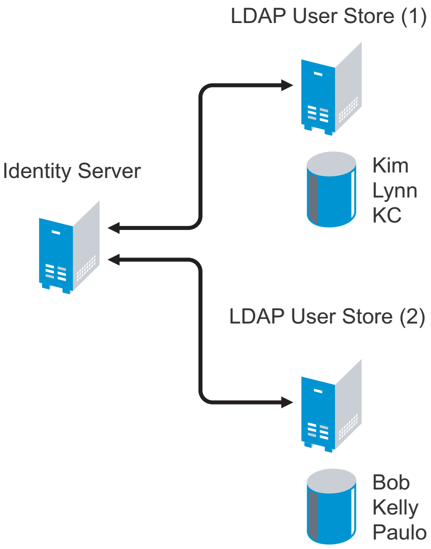
User stores are LDAP directory servers to which end users authenticate. You must specify an initial user store when creating an Identity Server configuration. This procedure describes how to add an additional user store to provide load balancing and failover capability. However, you use the same pages for setting up the initial user store, adding a user store, or modifying an existing user store.

- Section 8.1.1, "Using More Than One LDAP User Store," on page 90
- Section 8.1.2, "Configuring the User Store," on page 91
- Section 8.1.3, "Configuring an Admin User for the User Store," on page 93
- Section 8.1.4, "Configuring a User Store for Secrets," on page 94

8.1.1 Using More Than One LDAP User Store

You can also configure the Identity Server to search more than one user store during authentication. Figure 8-2 illustrates this type of configuration.

Figure 8-2 *Multiple LDAP Directories*



It is assumed that each LDAP directory contains different users. You should make sure the users have unique names across all LDAP directories. If both directories contain a user with an identical name, the name and password information discovered in the search of the first directory is always used for authentication. You select the user store and specify the search order when configuring the authentication method.

When users are added to the configuration store, objects are created for Access Manager profiles. If you delete a user from the LDAP directory, orphaned objects for that user remain in the configuration store. Ensure that you delete those objects as well.

If you add a secondary Administration Console and you have added replicas to the user store of the primary Administration Console, ensure that you also add the replicas to the secondary Administration Console.

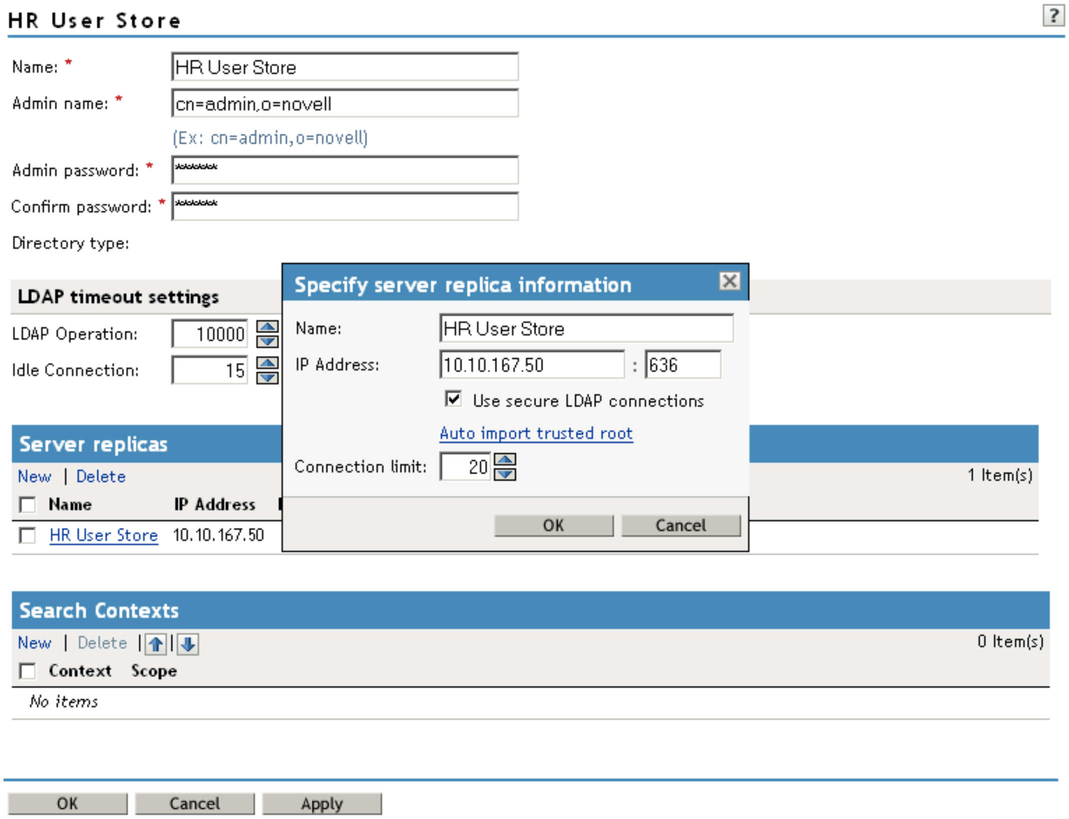
All user stores that you add are included in health checks. If health problems are found, the system displays the user store on the Health page and in the trace log file.

8.1.2 Configuring the User Store

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Local*.

2 In the User Stores list, click *New*.

If you are creating an Identity Server configuration, this is Step 3 of the wizard.



HR User Store

Name: * HR User Store

Admin name: * cn=admin,o=novell

(Ex: cn=admin,o=novell)

Admin password: * ******

Confirm password: * ******

Directory type:

LDAP timeout settings

LDAP Operation: 10000

Idle Connection: 15

Specify server replica information

Name: HR User Store

IP Address: 10.10.167.50 : 636

☑ Use secure LDAP connections

Auto import trusted root

Connection limit: 20

Server replicas

1 Item(s)

☐	Name	IP Address
☐	HR User Store	10.10.167.50

Search Contexts

0 Item(s)

☐	Context	Scope
	No items	

3 Fill in the following fields:

Name: The name of the user store for reference.

Admin Name: The distinguished name of the admin user of the LDAP directory, or a proxy user with specific LDAP rights to perform searches. Administrator-level rights are required for setting up a user store. This ensures read/write access to all objects used by Access Manager. For more information about this user, see Section 8.1.3, "Configuring an Admin User for the User Store," on page 93.

Each directory type uses a slightly different format for the DN:

- **eDirectory:** cn=admin,ou=users,o=novell
- **Active Directory:** cn=Administrator,cn=users,dc=domeh,dc=test,dc=com
- **Sun ONE:** cn=admin,cn=users,dc=novell,dc=com

Admin Password and Confirm Password: Specify the password for the admin user and confirm it.

Directory Type: The type of LDAP directory. You can select eDirectory, Active Directory, or Sun ONE.

If eDirectory has been configured to use Domain Services for Windows, eDirectory behaves like Active Directory. When you configure such a directory to be a user store, its Directory Type must be set to Active Directory for proper operation.

4 Under *LDAP timeout settings*, specify the following:

LDAP Operation: Specifies how long in seconds a transaction can take before timing out.

Idle Connection: Specifies how long in seconds before connections begin closing. If a connection has been idle for this amount of time, the system creates another connection.

5 Specify a server replica.

For an eDirectory™ server, it is recommended that you use a replica of the partition where the users reside. Ensure that each LDAP server in the cluster has a valid read/write replica. One option is to create a users partition (a partition that points to the OU containing the user accounts) and reference this server replica.

Name: The display name for the LDAP directory server. If your LDAP directory is replicated on multiple servers, use this name to identify a specific replica.

IP Address: The IP address of the LDAP directory server.

Port: The port of the LDAP directory server.

Use secure LDAP connections: (Required) Specifies that the LDAP directory server requires secure (SSL) connections with the Identity Server.

This option must be enabled if you use this user store as a Novell SecretStore™ User Store Reference in the Credential Profile details. (See Section 12.4, "Configuring Credential Profile Security and Display Settings," on page 176.) If you have specified that this user store is a SecretStore User Store Reference, this option is enabled but not editable.

Connection limit: The maximum number of pooled simultaneous connections allowed to the LDAP server. Valid values are between 5 and 100.

6 Click *Auto import trusted root*.

7 Click *OK* to confirm the import.

8 Select one of the certificates in the list.

You are prompted to choose either a server certificate or a root CA certificate. To trust one certificate, choose *Server Certificate*. Choose *Root CA Certificate* to trust any certificate signed by that certificate authority.

9 Specify an alias, then click *OK*.

10 Click *OK* in the *Specify server replica information* dialog box.

11 Select the replica, then click *Validate* to test the connection between the Identity Server and the replica.

The system displays the result under *Validation Status*. The system displays a green check mark if the connection is valid (the IP address and port) and the credentials (admin name and password). It does not check the certificate.

12 (Optional) To add additional replicas for the same user store, repeat Step 5 through Step 11.

For load balancing, a hash algorithm is used to map a user to a replica. All requests on behalf of that user are sent to that replica.

13 Add a Search Context.

The search context is used to locate users in the directory when a contract is executed.

- If a user exists outside of the specified search context (object, subtree, one level), the Identity Server cannot find the user, and the user cannot log in.
- If the search context is too broad, the Identity Server might find more than one match, in which case the contract fails, and the user cannot log in.

For example, if you allow users to have the same username and these users exist in the specified search context, these users cannot login if you are using a simple username and password contract. The search for users matching this contract will return more than one match. In this case, you need to create a contract that specifies additional attributes so that the search returns only one match. For more information on how to create such contracts, see Appendix F, "Authentication Classes and Duplicate Common Names," on page 719

A search context is required for Active Directory* or Sun ONE*. It is optional for eDirectory because the entire tree is searched from the root if a search context is not specified.

IMPORTANT: For Active Directory, do not set the search context at the root level by using the Subtree scope. This setting can cause serious performance problems. It is recommended that you set multiple search contexts, one for each top-level organizational unit.

14 Click *Finish*.

15 Add the new user store to the authentication method. See Section 8.3, "Configuring Authentication Methods," on page 109.

8.1.3 Configuring an Admin User for the User Store

The Identity Server must log in to each configured user store. It searches for users, and when a user is found, it reads the user's attributes values. When you configure a user store, you must supply the distinguished name of the user you want the Identity Server to use for logging in. You can use the admin user of your user store, or you can create a specialized admin user for the this purpose. When creating this admin user, you need to grant the following rights:

- The admin user needs rights to browse the tree, so the Identity Server can find the user who is trying to authenticate. The admin user needs browse rights to object class that defines the users and read and compare rights to the attributes of that class. When looking for the user, the Identity Server uses the GUID and naming attributes of the user class.

Directory	Object Class	GUID Attribute	Naming Attribute
eDirectory	User	guid	cn
Active Directory	User	objectGUID	sAMAccountName
Sun One	inetOrgPerson	nsuniqueid	uid

- The admin user needs read rights to any attributes used in policies (Role, Form Fill, Identity Injection, Authorization).
- If a secret store is used in Form Fill policies, the admin user needs write rights to the attributes storing the secrets.

- If a password management servlet is enabled, the admin user needs read rights to the attributes controlling grace login limits and remaining grace logins.
- If you enable provisioning with the SAML or Liberty protocols, the admin user needs write rights to create users in the user store.

8.1.4 Configuring a User Store for Secrets

Access Manager allows you to securely store user secrets. These secrets can then be used in Form Fill and Identity Injection policies. Where and how the secrets are stored depends upon your user store and your configuration:

- "Configuring the Configuration Datastore to Store the Secrets" on page 94.

  If you want to do minimal configuration, you can use the configuration datastore on the Administration Console to store the secrets. To increase the security of the secrets, you should configure the security options.
- "Configuring an LDAP Directory to Store the Secrets" on page 96.

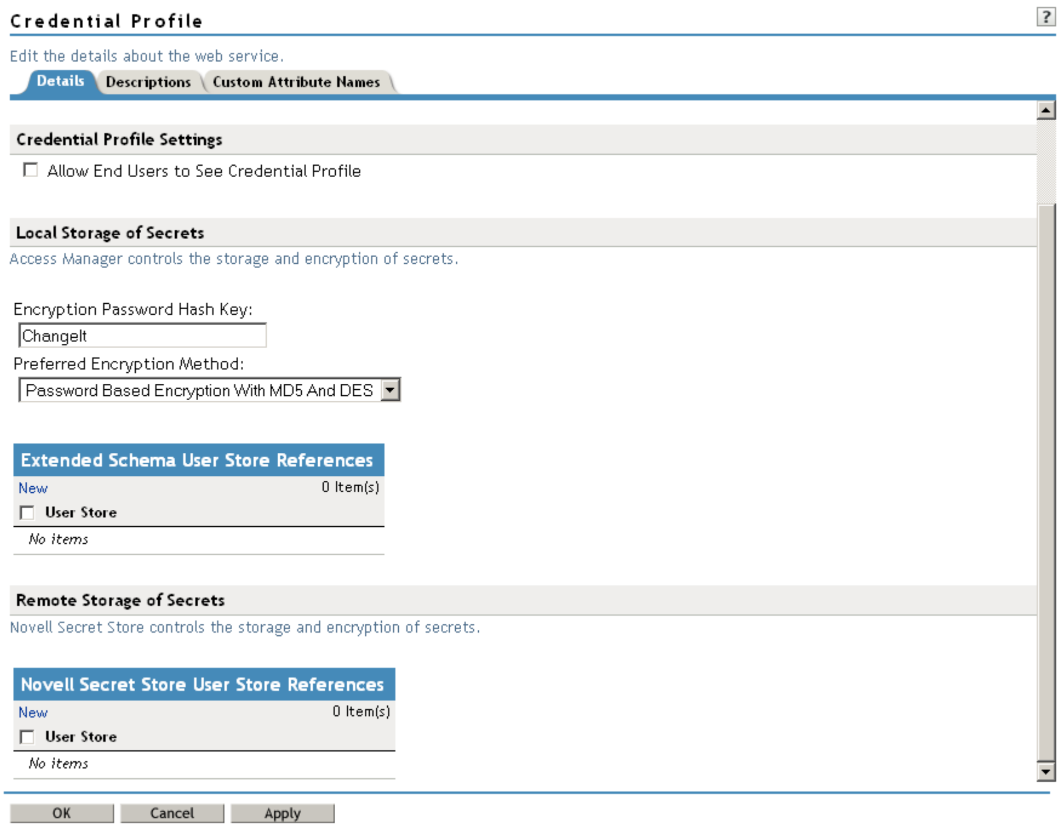
  If you are willing to extend the schema and add an attribute to your user object on the LDAP directory, you can store the secrets in your LDAP directory.
- "Configuring an eDirectory User Store to Use SecretStore" on page 97.

  If your user store is eDirectory and you have installed Novell SecretStore®, you can select to use the SecretStore on your eDirectory server to store the secrets.

Configuring the Configuration Datastore to Store the Secrets

When you use the configuration datastore of the Administration Console as the secret store, the nidswsfss attribute of the nidsLibertyUserProfile object is used to store the secrets.

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Providers*.
2. Click *Credential Profile*.



Edit the details about the web service.

Credential Profile Settings

☐ Allow End Users to See Credential Profile

Local Storage of Secrets

Access Manager controls the storage and encryption of secrets.

Encryption Password Hash Key: ChangeIt

Preferred Encryption Method: Password Based Encryption With MD5 And DES

Extended Schema User Store References

No items

Remote Storage of Secrets

Novell Secret Store controls the storage and encryption of secrets.

Novell Secret Store User Store References

No items

3 Scroll to the *Local Storage of Secrets* section and configure the following security options:

Encryption Password Hash Key: (Required) Specifies the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, we recommend that you change the default password to a unique alphanumeric value.

Preferred Encryption Method: Specifies the preferred encryption method. Select the method that complies with your security model:

- **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key.
- **DES:** Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
- **Triple DES:** A variant of DES in which data is encrypted three times with standard DES, using two different keys.

Extended Schema User Store References: Do not specify a user store reference. When this option contains no values, the configuration datastore is used to store the secrets.

4 Click *OK*.

5 On the Identity Servers page, update the Identity Server.

6 To use the secret store to store policy secrets, see Section 30.4, “Creating and Managing Shared Secrets,” on page 499.

Configuring an LDAP Directory to Store the Secrets

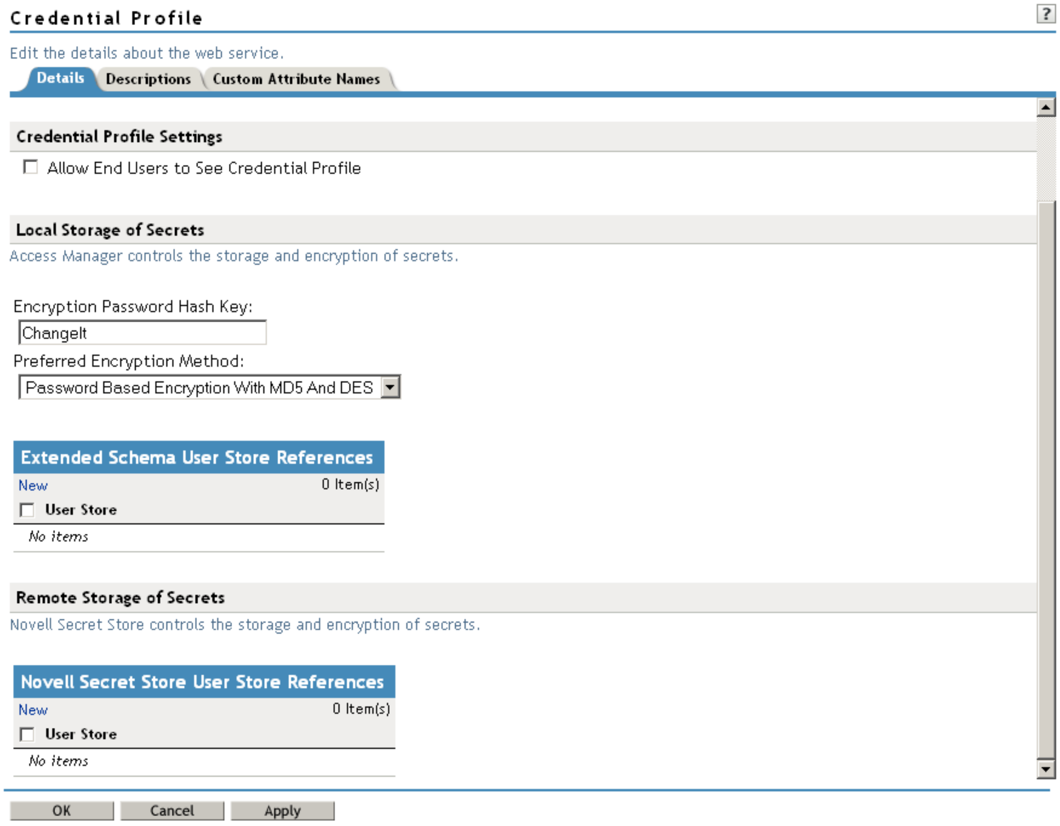
When you use an LDAP directory to store the secrets, you need to enable the user store for the secrets. You select the LDAP directory, then specify an attribute. The attribute you specify is used to store an XML document that contains encrypted secret values. This attribute should be a single-valued case ignore string that you have defined and assigned to the user object in the schema.

To use an LDAP directory to store secrets, your network environment must conform to the following requirements:

- The user class object must contain an attribute that can be used to store the secrets. This attribute must be a string attribute that is single valued and case ignore.
- The user store must be configured to use secure connections (click *Access Manager* > *Identity Servers* > *Edit* > *Local* > *User Stores* > *[User Store Name]*. In the *Server replicas* section, ensure that the *Port* is 636 and that *Use SSL* is enabled. If they aren't, click the name of the replica and reconfigure it.

To configure the LDAP directory:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty* > *Web Service Providers*.
2. Click *Credential Profile*.



Edit the details about the web service.

Credential Profile Settings

☐ Allow End Users to See Credential Profile

Local Storage of Secrets

Access Manager controls the storage and encryption of secrets.

Encryption Password Hash Key: ChangeIt

Preferred Encryption Method: Password Based Encryption With MD5 And DES

Extended Schema User Store References

New 0 Item(s)

☐ User Store

No items

Remote Storage of Secrets

Novell Secret Store controls the storage and encryption of secrets.

Novell Secret Store User Store References

New 0 Item(s)

☐ User Store

No items

3. Scroll to the *Local Storage of Secrets* section and configure the following options:

Encryption Password Hash Key: (Required) Specifies the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, we recommend that you change the default password to a unique alphanumeric value.

Preferred Encryption Method: Specifies the preferred encryption method. Select the method that complies with your security model:

- **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key.
- **DES:** Data Encryption Standard (DES) is a widely used method of data encryption that uses a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
- **Triple DES:** A variant of DES in which data is encrypted three times with standard DES, using two different keys.

4. To specify where to store secret data, click *New* under *Extended Schema User Store References* and fill in the following:

   User Store: Select the user store where you want secret store enabled.

   Attribute Name: Specify the LDAP attribute that you have created to store the secrets on the selected user store.

5. Click *OK* twice.
6. On the Identity Servers page, update the Identity Server.
7. To create policies that use the stored secrets, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.

For troubleshooting information, see "Troubleshooting the Storing of Secrets" on page 99.

Configuring an eDirectory User Store to Use SecretStore

For Access Manager to use Novell SecretStore, the user store must be eDirectory and Novell SecretStore must be installed there. When configuring this user store for secrets, Access Manager extends the eDirectory schema for an NMAS™ method. This method converts authentication credentials to a form understood by eDirectory. For example, Access Manager supports smart card and token authentications, and these authentication credentials must be converted into the username and password credentials that eDirectory requires. This allows the Identity Server to authenticate as that user and access the user's secrets. Without this NMAS method, the Identity Server is denied access to the user's secrets.

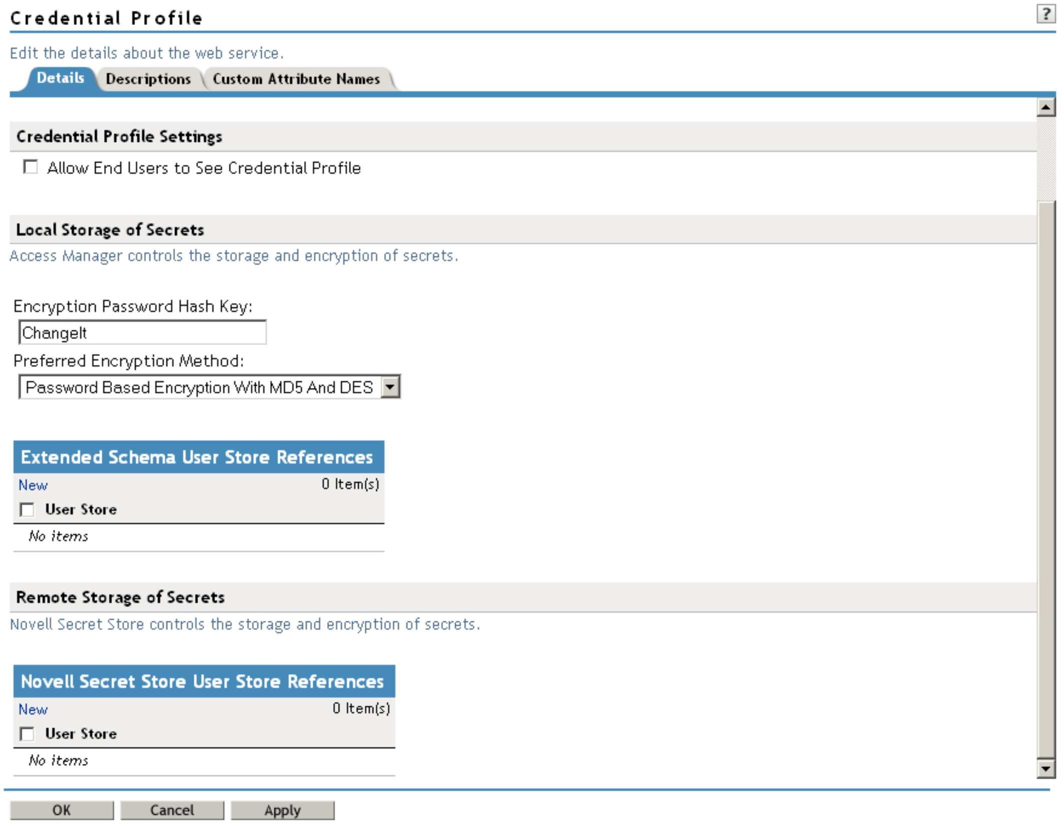
To use a remote SecretStore, your network environment must conform to the following requirements:

- The eDirectory server must have Novell SecretStore installed.
- When you configure a user store to use Novell SecretStore, the admin user (see Section 8.1.3, "Configuring an Admin User for the User Store," on page 93) you have configured for the user store must have sufficient rights to extend the schema on the eDirectory server, to install the SAML NMAS method, and set up the required certificates and objects.
- The user store must be configured to use secure connections (click *Access Manager > Identity Servers > Edit > Local > User Stores > [User Store Name]*. In the *Server replicas* section, ensure that the *Port* is 636 and that *Use SSL* is enabled. If they aren't, click the name of the replica and reconfigure it.

- If you have enabled a firewall between the Administration Console and the user store, and between the Identity Server and the user store, make sure that both LDAP ports (389 and 636) and the NCP™ port (524) are opened.
- If you are going to configure Access Manager to use secrets that are used by other applications, you need to plan a configuration that allows the user to unlock a locked SecretStore. See "Determining a Strategy for Unlocking the SecretStore" on page 99.

To configure the user store:

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Providers*.
2. Click *Credential Profile*.



Credential Profile

Edit the details about the web service.

Credential Profile Settings

☐ Allow End Users to See Credential Profile

Local Storage of Secrets

Access Manager controls the storage and encryption of secrets.

Encryption Password Hash Key: ChangeIt

Preferred Encryption Method: Password Based Encryption With MD5 And DES

Extended Schema User Store References

New 0 Item(s)

☐ User Store

No items

Remote Storage of Secrets

Novell Secret Store controls the storage and encryption of secrets.

Novell Secret Store User Store References

New 0 Item(s)

☐ User Store

No items

3. Scroll to the *Remote Storage of Secrets* section.
4. Click *New* under *Novell Secret Store User Store References*.

   This adds a reference to a user store where SecretStore has been installed.
5. Click the user store that you configured for SecretStore.
6. Click *OK* twice.
7. On the Identity Servers page, update the Identity Server.
8. Continue with one of the following:
   - If other applications are using the secret store, you need to determine whether Access Manager users need the option to unlock the secret store. See "Determining a Strategy for Unlocking the SecretStore" on page 99.

- To create policies that use the stored secrets, see "Creating and Managing Shared Secrets" on page 499.
- For troubleshooting information, see "Troubleshooting the Storing of Secrets" on page 99.

Determining a Strategy for Unlocking the SecretStore

When an administrator resets a user's password, secrets written to the Novell SecretStore with an enhanced security flag become locked. The Identity Server does not write the secrets that it creates with this flag, but other applications might:

- If Access Manager is not sharing secrets with other applications, the secrets it is using are never locked, and you do not need to configure Access Manager to unlock secrets.
- If Access Manager is sharing secrets with other applications and these application are using the security flag that locks secrets when a user's password is reset, you need to configure Access Manager so that users can unlock their secrets.

If you want users to receive a prompt to for their passphrase when secrets are locked, complete the following configuration steps:

1 At the Identity Server, configure Tomcat to perform a check to verify if the SecretStore is locked:

  1a In a text editor, open the following file:

  ```
  /var/opt/novell/tomcat4/conf/tomcat4.conf
  ```

  1b Add the following command to the end of the file and save your changes:

  ```
  JAVA_OPTS=${JAVA_OPTS} -Dnids.sscheck=true
  ```

  1c Restart Tomcat by using the following command:

  ```
  /etc/init.d/novell-tomcat4 restart
  ```

2 Make sure Web Services Framework is enabled:

  2a In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Liberty* > *Web Services Framework*.

  2b In the *Framework General Settings* section, make sure that *Enable Framework* is selected.

  2c Click *OK*. If you made any changes, update the Identity Server.

3 Continue with "Creating and Managing Shared Secrets" on page 499.

When the SecretStore is locked and the users log in, the users are first prompted for their login credentials, then prompted for the passphrase that is used to unlock the SecretStore.

Troubleshooting the Storing of Secrets

- "Secrets Aren't Stored in Novell SecretStore" on page 99
- "Users Are Receiving Invalid Credential Messages" on page 101
- "Secrets Aren't Stored in the LDAP Directory" on page 101

Secrets Aren't Stored in Novell SecretStore

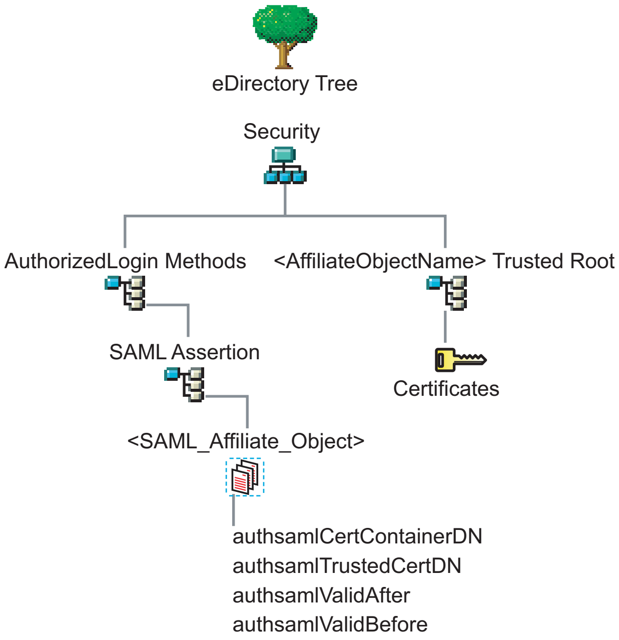
When you use Novell SecretStore to store the secrets, the schema on the eDirectory server must be extended, and specific SAML objects and certificates must be created.

To verify that the schema was extended and the objects were created on the eDirectory server:

1 Open an LDAP browser and connect to the eDirectory server.

2 Browse to the Security container.

3 Look for objects similar to the following:



If the schema has been extended correctly, you can find a SAML Assertion object in the Authorized Login Methods container. The SAML_Assertion object contains an alphanumeric generated name for a SAML affiliate object. This object has four attributes.

The SAML affiliate object name is used to generate another container in the Security container. This new container is the <*AffiliateObjectName*> Trusted Root container that contains public key signing certificate.

4 Complete one of the following:

- If these objects do not exist, verify the following, then continue with Step 5:
  - The admin user for the user store has sufficient rights to extend the schema and add these objects to the Security container.
  - Any configured firewalls must allow NCP and LDAP traffic for the Administration Console, the Identity Server, and the LDAP user store.
- If the objects exist, check for time synchronization problems. For more information, see "Users Are Receiving Invalid Credential Messages" on page 101.

5 In the Administration Console, modify the secret store configuration so that it is resent to the user store:

5a Click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty* > *Web Service Providers* > *Credential Profile*.

5b In the *Remote Storage of Secrets* section, remove the user store, then add it again.

5c Click *OK*.

6 On the Identity Servers page, update the Identity Server.

Users Are Receiving Invalid Credential Messages

The <SAML_Affiliate_Object>.SAML-Assertion.AuthorizedLoginMethods.Security object contains two attributes that determine how long credentials are valid. If your Identity Server and eDirectory server are not time synchronized, the credentials can become invalid before a user has time to use them.

Either make sure that the time of your Identity Server and eDirectory server are synchronized, or increase the value of the authsamlValidAfter and authsamlValidBefore attributes of the SAML affiliate object.

Secrets Aren't Stored in the LDAP Directory

1. Open an LDAP browser and connect to the eDirectory server.
2. Browse to the user object.
3. Verify that the user object contains the LDAP attribute that you have specified as the attribute to store the secrets.
4. If the attribute exists, browse to the schema and verify that the attribute has the following characteristics:
    - Single valued
    - Case ignore
    - String

8.2 Creating Authentication Classes

Authentication classes let you define ways of obtaining end user credentials.You specify the code (Java class) and properties to be executed to implement a particular authentication type.

Several authentication classes are included with Access Manager to provide a variety of ways to authenticate end users. Custom authentication classes provided by other vendors can also be configured to run in the system.

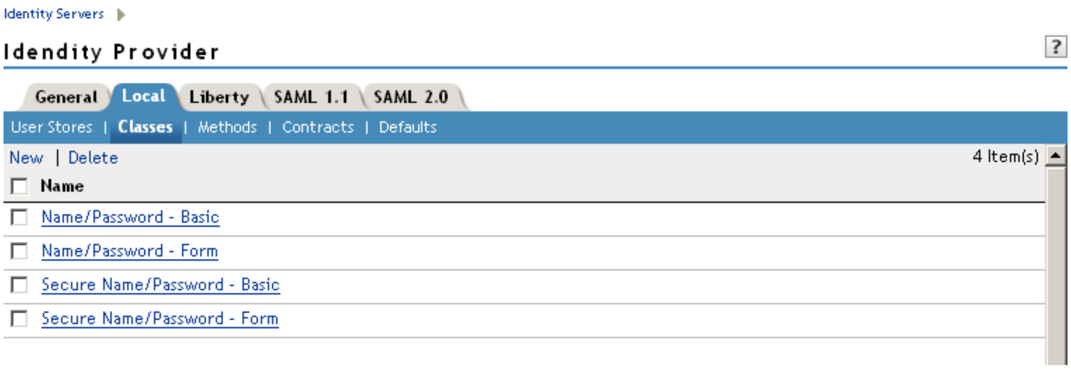
- Section 8.2.1, "Creating Basic, Form-Based, or NMAS Authentication Classes," on page 101
- Section 8.2.2, "Creating an X.509 Authentication Class," on page 104
- Section 8.2.3, "Creating a RADIUS Authentication Class," on page 108

Some classes require additional configuration to enable their use for authentication. See the following sections:

- Section 8.9, "Configuring Kerberos for Authentication," on page 119
- Section 8.10, "Configuring Access Manager for NESCM," on page 130

8.2.1 Creating Basic, Form-Based, or NMAS Authentication Classes

1. In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Local > Classes*.



☐ Name
☐ Name/Password - Basic
☐ Name/Password - Form
☐ Secure Name/Password - Basic
☐ Secure Name/Password - Form

The following classes are predefined for Access Manager:

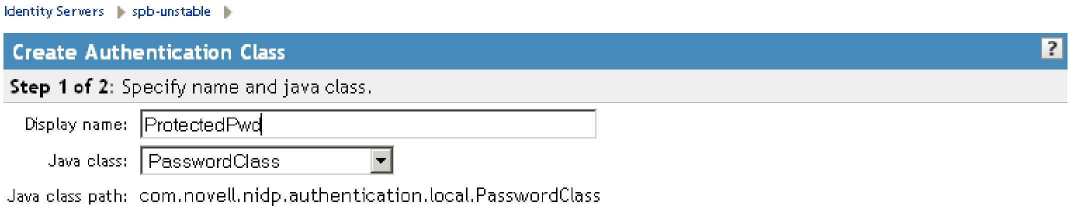
Name/Password - Basic: Basic authentication over HTTP using a standard login pop-up page provided by the Web browser.

Name/Password - Form: Form-based authentication over HTTP.

Secure Name/Password - Basic: Basic authentication over HTTPS using a standard login page provided by the Web browser.

Secure Name/Password - Form: Form-based authentication over HTTPS.

2 Click *New* to launch the Create Authentication Class Wizard.



Step 1 of 2: Specify name and java class.

Display name: ProtectedPwd

Java class: PasswordClass

Java class path: com.novell.nidp.authentication.local.PasswordClass

3 Specify a display name, then select a class from the *Java class* drop-down menu.

The following classes are recommended only for testing purposes:

BasicClass: Uses basic HTTP authentication.

PasswordClass: Passes the user name and password over HTTP in readable text, and uses a form-based login to collect the name and password.

RadiusClass: RADIUS enables communication between remote access servers and a central server. For a production environment, use ProtectedRadiusClass. See Section 8.2.3, “Creating a RADIUS Authentication Class,” on page 108 for configuration steps.

For a production environment, select one of the following protected classes:

X509Class: See Section 8.2.2, “Creating an X.509 Authentication Class,” on page 104.

ProtectedBasicClass: The BasicClass, protected by HTTPS.

ProtectedPasswordClass: The PasswordClass, protected by HTTPS (form-based).

ProtectedRadiusClass: The RadiusClass, protected by HTTPS. See Section 8.2.3, “Creating a RADIUS Authentication Class,” on page 108 for configuration steps.

NMASAuthClass: The authentication class used for Novell Modular Authentication Service (NMAS™), which uses fingerprint and other technology as a means to authenticate a user. For instructions on using the NMAS NESCM method, see Section 8.10, “Configuring Access Manager for NESCM,” on page 130.

KerberosClass: The authentication class used for using Kerberos for Active Directory and Identity Server authentication. See Section 8.9, “Configuring Kerberos for Authentication,” on page 119 for configuration steps.

Other: Used for third-party authentication classes or if you have written your own Java class. For information on how to write your own class, see Novell Access Manager Developer Tools and Examples (http://developer.novell.com/wiki/index.php/Novell_Access_Manager_Developer_Tools_and_Examples).

To download an authentication class that retrieves the user’s password and injects it into the user’s credentials when the user authenticates using a non-password method such as X509, Radius, smart card, or Kerberos, see Access Management Authentication Class Extension to Retrieve Password for Single Sign-on (http://www.novell.com/communities/node/4556). Such a class allows you to enable single sign-on with Identity Injection and Form Fill policies that require the user’s password.

4 Click *Next* to configure the properties for each class. The values you enter are case sensitive.

Property Name	Class	Property Value
Query	BasicClass PasswordClass ProtectedBasicClass ProtectedPasswordClass	As an example, if you specify the property value of *(&(objectclass=person)(email=%EMail Value%))*, the %EMail Value% is replaced with the name entered in the basic authentication login. The property value of the Query must be a valid LDAP query string.
<filename> This property name must be the name of a new JSP™ file that includes all the needed fields for the Query property. The property value of this attribute should not include the `.jsp` extension of the file. For example, if you create a new JSP file named `login2.jsp` then the value of the property would be `login2`.	ProtectedBasicClass ProtectedPasswordClass	The property value for JSP is the name of the JSP page you customized. For example, if you use *(&(objectclass=person)(cn=%Ecom_User_ID%)(mail=%Ecom_Email%))* as the Query property value, the system queries for an object of type *person* that contains a cn equal to *Ecom_User_ID* from the specified `.jsp` file, and mail equal to *Ecom_Email* from the same `.jsp` file. The `.jsp` file for this type of query must prompt the user for a username and an email address. See Section 8.7, “Creating Custom Login Pages,” on page 114 for more information.

Property Name	Class	Property Value
NMAS_LOGIN_SEQUENCE	NMASAuthClass	Specify the name of the NMAS Login Sequence to be used for this type of authentication. See Section 8.10, "Configuring Access Manager for NESCM," on page 130.

5 Click *Finish*.

6 Continue with Section 8.3, "Configuring Authentication Methods," on page 109.

To use an authentication class, the class must have one or more associated methods.

8.2.2 Creating an X.509 Authentication Class

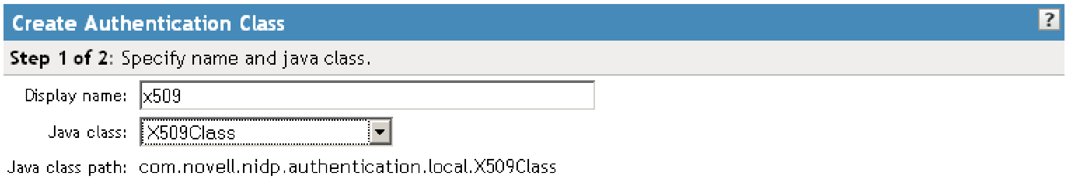
The X.509 authentication class lets you authenticate users using X.509 certification for mutual authentication. It also identifies the user in user-stores, employing various user-mapping mechanisms.

IMPORTANT: If you have installed your Identity Server as a protected Access Gateway resource, you cannot use the x.509 class for authentication. For more information about this type of configuration, see "Protecting an Identity Server with an Access Gateway" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

1 In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Local* > *Classes*.

2 Click *New*.

3 Specify a display name, then choose X509Class from the drop-down menu.



Create Authentication Class

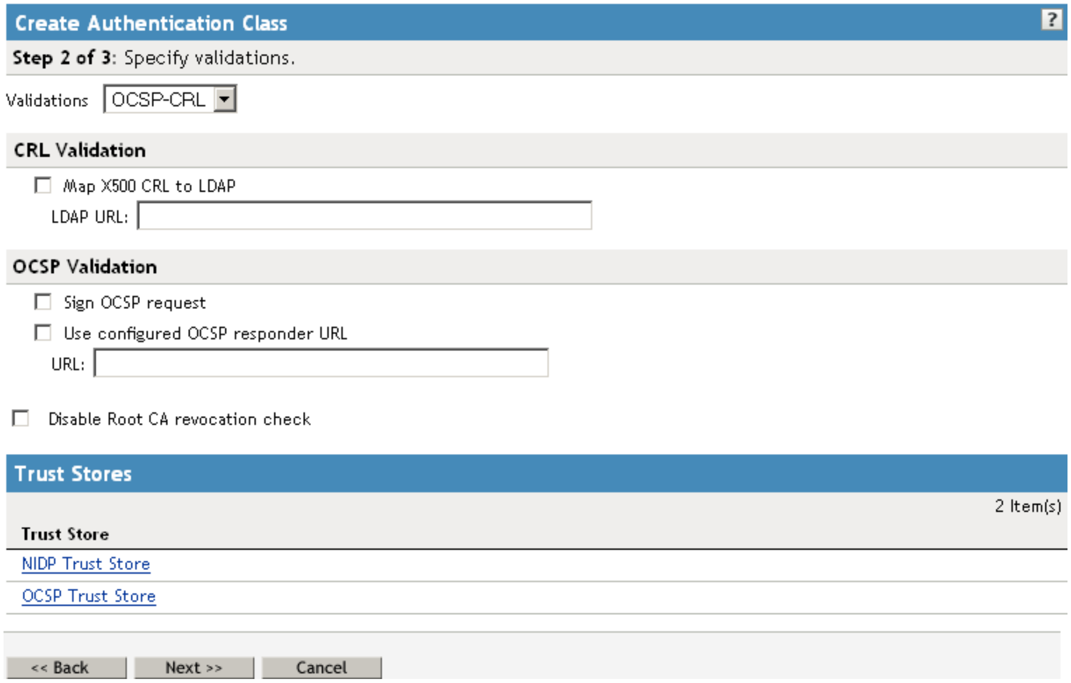
Step 1 of 2: Specify name and java class.

Display name: x509

Java class: X509Class

Java class path: com.novell.nidp.authentication.local.X509Class

4 Click *Next*.



Create Authentication Class

Step 2 of 3: Specify validations.

Validations: OCSP-CRL

CRL Validation

☐ Map X500 CRL to LDAP

LDAP URL:

OCSP Validation

☐ Sign OCSP request

☐ Use configured OCSP responder URL

URL:

☐ Disable Root CA revocation check

Trust Stores

2 Item(s)

Trust Store
NIDP Trust Store
OCSP Trust Store

5 Configure the validation options:

Validations: The validation type. Trust validation occurs if the certificate chain is verified in the *NIDP Trust Store*. In addition to usual certificate validations, the Identity Server supports CRL (certificate revocation list) and OCSP (Online Certificate Status Protocol) validations for each authentication request.

Access Manager caches CRLs, so the revoked status of a newly revoked certificate is not picked up until the next cache refresh. For higher security requirements, use OCSP validation with CRL validation. You can select None, CRL, OCSP, OCSP-CRL, or CRL-OCSP validation. In a production environment, for highest security, select either OCSP-CRL or CRL-OCSP validation. The default setting is to check OCSP first, then CRL.

CRL Validation: Checks the CRL. If you enable CRL validations, the CRL distribution point extension is read out of the user's X.509 certificate. The CRL distribution point contains URL where the complete CRL can be found, as published by the certificate authority. The system performs sanity checks on the CRL itself and then checks to see if the user certificate is in the revoked list. The system can get the CRL over HTTP and LDAP. If you are not expecting the distribution point in user certificates, you can specify a value in the *LDAP URL* to get the CRL.

Access Manager supports two schemes for a URL: `http://` and `ldap://`.

The CRL is cached until Tomcat is restarted.

OCSP Validation: If OCSP validation is enabled, the Authority Info Access point (AIA) is read out of the user certificate, which contains the URL for the OCSP responder. A signed OCSP request for the user certificate is sent to OCSP responder. A signed OCSP response is received from the responder which has the revoked status for the user certificate. Alternately, if you are not expecting AIA in user certificate, you can specify a value in the OCSP responder *URL* field. The value you enter here overrides any OCSP responder URLs in a certificate.

Access Manager supports two schemes for a URL: `http://` and `ldap://`.

Disable Root CA Revocation Check: Disables whether to check if a certificate authority has been revoked. This option checks the CRL and OCSP for the trusted root certificate in the chain. You can enable or disable this option for X.509 user authentication performance.

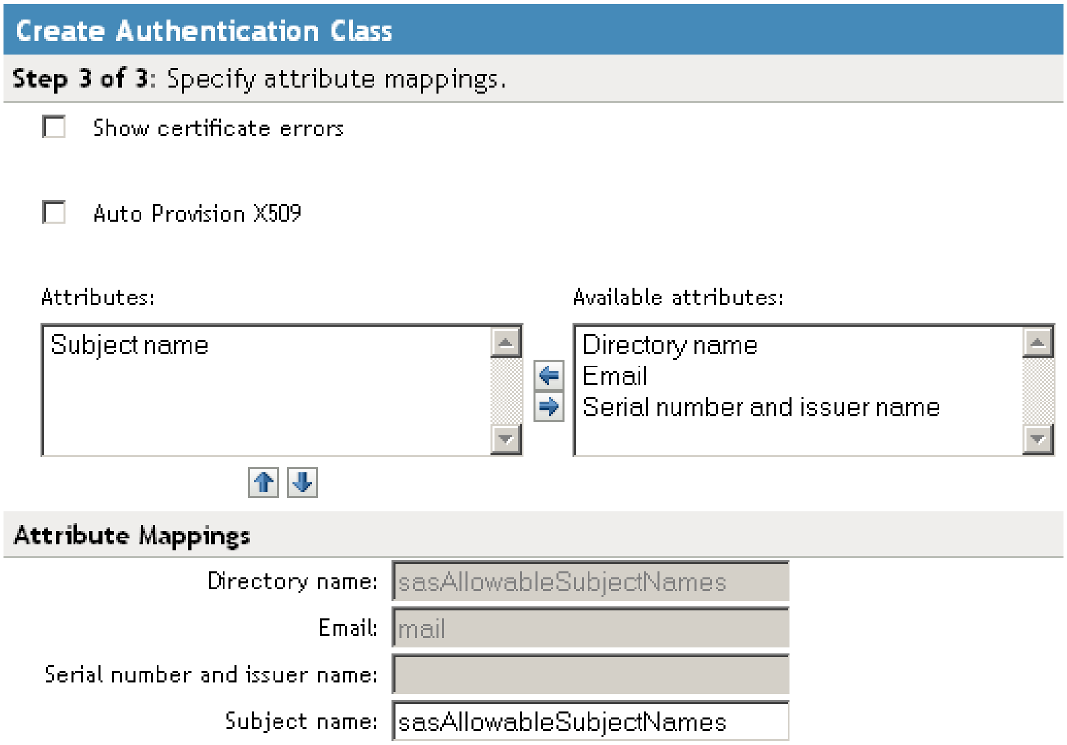
6 Configure the trust stores:

NIDP Trust Store: This trust store must contain the trusted root certificate of the Certificate Authorities that signed your user certificates. Click this link to add certificates to the trust store.

OCSP Trust Store: This trust store must contain the signing certificate of the OCSP servers you want to trust. Click this link to add certificates to the trust store. You must add the signing certificate, not the trusted root certificate, for this feature to work.

7 Click *Next*.

8 Configure attribute mappings.



Create Authentication Class

Step 3 of 3: Specify attribute mappings.

☐ Show certificate errors

☐ Auto Provision X509

Attributes:

Subject name

Available attributes:

Directory name
Email
Serial number and issuer name

Attribute Mappings

Directory name:	sasAllowableSubjectNames
Email:	mail
Serial number and issuer name:	
Subject name:	sasAllowableSubjectNames

Use this page to specify attribute mappings for the X.509 authentication class. *Subject name* is the default map.

Show certificate errors: Displays an error page when a certificate error occurs. This option is disabled by default. When troubleshooting mutual authentication, you should enable this option. It is a good way to gather additional information.

Auto Provision X509: Enables automatic provisioning of users using X.509 authentication. This option allows you to activate X.509 for increased security, while using a less secure way of authentication, such as username/password. Extra security measures can even include manual intervention to activate X.509 authentication by adding an extra attribute that is checked during authentication.

An example of using this option is when a user authenticates with an X.509 certificate, a lookup is performed for a matching SASallowableSubjectNames with the name of the user certificate. When no match is found, and *Auto Provision X509* is enabled, the user is presented with a custom error page specifying to click a button provide additional credentials, such as a username and password, or to start an optional Identity Manager workflow. If the authentication is successful, then the user's SASallowableSubjectNames attribute is filled in with the certificate name of the user certificate.

When *Auto Provision X509* is enabled, and the attribute that is used for subject name mapping is changed from the default sasAllowableSubjectNames, ensure that the LDAP attribute that is used can store string values with a length as long as the longest client certificate subject name. For example, if you use the LDAP attribute title (which has an upper bound of 64 characters) the *Auto Provision X509* fails the provisioning part of the authentication, if the client certificate subject name is longer 64 characters. The authentication works if a valid name and password is given. However, provisioning fails.

Attributes: The list of attributes currently used for matching. If multiple attributes are added to the list, the user must match only one of the attributes. As soon as a match occurs, the other attributes are skipped. Use the Up and Down arrows to arrange the order in which the system processes the attributes.

Available attributes: The available X.509 attributes. To use an attribute, select it and move it to the *Attributes* field. When the attribute is moved to the *Attributes* list, you can modify the mapping name in the *Attribute Mappings* section. The mapped name must match an attribute in your LDAP user store.

Directory name: Searches for the Directory Address in the client certificate and tries to match it to the DN of a user in the user store. If that fails, it searches the sasAllowableSubjectNames attribute of all users for a value that matches. The sasAllowableSubjectNames attribute must contain values that are comma-delimited, with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

Email: Searches for the email attribute in the client certificate and tries to match it with a value in the LDAP *mail* attribute.

Serial number and issuer name: Lets you match a user's certificate by using the serial number and issuer name. The issuer name and the serial number must be put into the same LDAP attribute of the user, and the name of this attribute must be listed in the *Attribute Mappings* section.

When using a Case Ignore String attribute, both the issuer name and the serial number must be in the same attribute separated by a dollar sign ($) character. The issuer name must be in front of the $ character, with the serial number following the $ character. Do not use any spaces in front of or behind the $ character. (For example, O=CURLY, OU=Organization CA$021C0562C5C4...) The issuer name can be from root to leaf or from leaf to root. The issuer name must be comma-delimited with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

The LDAP attribute can be any Case Ignore List or Case Ignore String attribute of the user. If you are configuring your own attribute, ensure that the attribute is added to the Person class. When using a Case Ignore List attribute, both the issuer name and the serial number must be in the same list. The issuer name needs to be the first item in the list, with the serial number being the second and last item in the list.

The certificate number is displayed in Internet Explorer with a space after every fourth digit. However, you should enter the certificate number without using spaces.

Subject name: Searches for the Subject name of the client certificate and tries to match it to the DN of a user in the user store. If that fails, it searches the sasAllowableSubjectNames attribute of all users for a value that matches the Subject name of the client certificate. The sasAllowableSubjectNames attribute must contain values that are comma-delimited, with a space after the comma. (For example, O=CURLY, OU=Organization CA or OU=Organization CA, O=CURLY.)

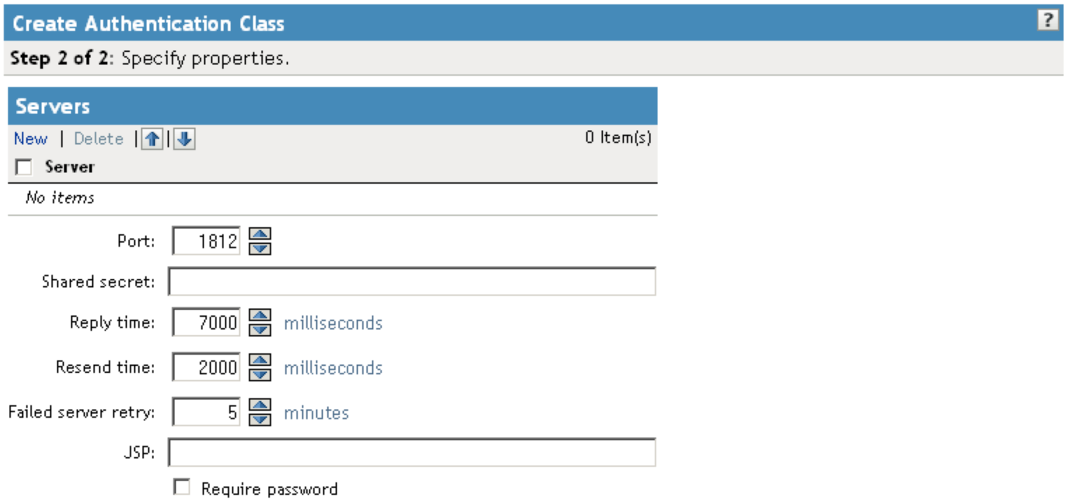
9. Click *Finish*.
10. Continue with Section 8.3, “Configuring Authentication Methods,” on page 109.

    To use an authentication class, the class must have one or more associated methods.

8.2.3 Creating a RADIUS Authentication Class

RADIUS enables communication between remote access servers and a central server. Secure token authentication through RADIUS is possible because Access Manager works with Novell Modular Authentication Service (NMAS) RADIUS software that can run on an existing NetWare® server. Access Manager supports both PIN and challenge and response methods of token-based authentication. In other words, RADIUS represents token-based authentication methods used to authenticate a user, based on something the user possesses (for example, a token card). Token challenge-response is supported for two-step processes that are necessary to authenticate a user.

1. In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Local* > *Classes*.
2. Click *New*.
3. Specify a display name, then select *RadiusClass* or *ProtectedRadiusClass* from the drop-down menu.
4. Click *Next*.



Create Authentication Class

Step 2 of 2: Specify properties.

Servers

Port: 1812

Shared secret:

Reply time: 7000 milliseconds

Resend time: 2000 milliseconds

Failed server retry: 5 minutes

JSP:

☐ Require password

5. Click *New* to add an IP address for the RADIUS server. You can add additional servers for failover purposes.
6. Click *OK*.
7. Fill in the following fields:

   Port: The port of the RADIUS server.

   Shared Secret: The RADIUS shared secret.

   Reply Time: The total time to wait for a reply in milliseconds

   Resend Time: The time to wait in milliseconds between requests.

Server Failure Retry: The time in milliseconds that must elapse before a failed server is retried.

JSP: The Java Server Page for the Java program executed by the Web server. Specify the name of the Java Server Page if you want to use something other than the provided JSP. The default page is used if nothing is specified.

- **Require Password:** Specifies whether to require a JSP password.

8 Click *Finish*.

9 Continue with Section 8.3, "Configuring Authentication Methods," on page 109.

To use an authentication class, the class must have one or more associated methods.

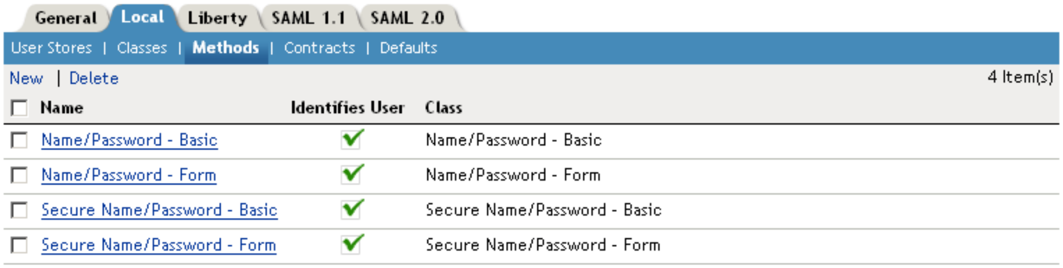
8.3 Configuring Authentication Methods

Authentication methods let you associate authentication classes with user stores. You use a particular authentication class to obtain credentials about an entity, and then validate those credentials against a list of user stores.

After the system locates the entity in a particular user store, no further checking occurs, even if the credentials fail to validate the entity. Typically, the entity being authenticated is a user, and the definition of an authentication method specifies whether this is the case. You can alter the behavior of an authentication class by specifying properties (name/value pairs) that override those of the authentication class.

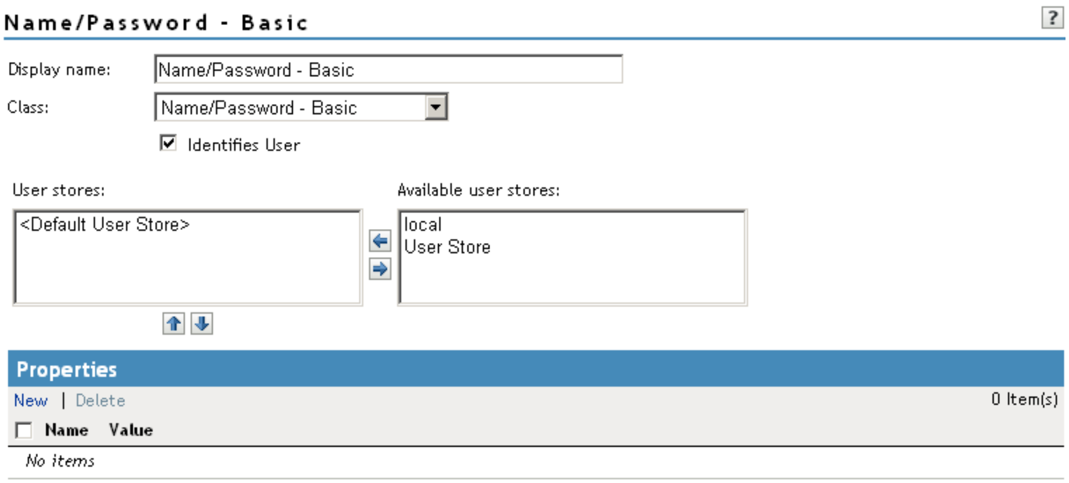
To configure a method for an authentication class:

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Local* > *Methods.*



☐ Name	Identifies User	Class
☐ Name/Password - Basic	✓	Name/Password - Basic
☐ Name/Password - Form	✓	Name/Password - Form
☐ Secure Name/Password - Basic	✓	Secure Name/Password - Basic
☐ Secure Name/Password - Form	✓	Secure Name/Password - Form

2 Click one of the predefined authentication methods, or click *New* to create one.



Name/Password - Basic

Display name: Name/Password - Basic

Class: Name/Password - Basic

☑ Identifies User

User stores: <Default User Store>

Available user stores: local, User Store

Properties

New | Delete 0 Item(s)

☐ Name	Value
No items	

3 Fill in the following fields:

Display Name: The name to be used to refer to the new method.

Class: The authentication class to use for this method. See Section 8.2, “Creating Authentication Classes,” on page 101.

Identifies User: Resolves to a user in the directory when credentials are provided. If this is not enabled, only the authentication is validated, such as the authentication of a computer. If multiple methods identify the user during the user session, all methods that identify the user must identify the same user in order for authentication to succeed.

4 Add user stores to search.

You can select from the list of all the user stores you have set up. If you have several user stores, the system searches through them based on the order specified here. If a user store is not moved to the *User stores* list, users in that user store cannot use this method for authentication.

<Default User Store>: The default user store in your system. See Section 8.5, “Specifying Authentication Defaults,” on page 113.

5 (Optional) Under Properties, click *New*, then fill in the following fields:

Property Name: The name of the property to be set. This value is case sensitive and specific to an authentication class. The same properties that can be set on an authentication class can be set on the method. For a list, see Step 4 in Section 8.2.1, “Creating Basic, Form-Based, or NMAS Authentication Classes,” on page 101.

You can use the method properties to override the property settings specified on the authentication class. For example, you might want to use the authentication class for multiple companies, but use a slightly different login page that is customized with the company’s logo. You can use the same authentication class, create a different method for each company, and use the filename property to specify the appropriate login page for each company.

The Radius classes have the following additional properties that can be set on the method:

- **RADIUS_LOOKUP_ATTR:** Defines an LDAP attribute whose value is read and used as the ID is passed to the RADIUS server. If not specified, the user name entered is used.
- **NAS_IP_ADDRESS:** Specifies an IP address used as a RADIUS attribute. You might use this property for situations in which service providers are using a cluster of small network access servers (NASs). The value you enter is sent to the RADIUS server.

Property Value: The values associated with the *Property Name* field.

6 Click *Finish*.

7 Continue with Section 8.4, “Configuring Authentication Contracts,” on page 111.

To use a method for authenticating a user, each method must have an associated contract. Contracts are assigned to resources, and it is access to a resource that triggers the authentication process. If the user has already supplied the required credentials for the contract, the user is not prompted for them again.

8.4 Configuring Authentication Contracts

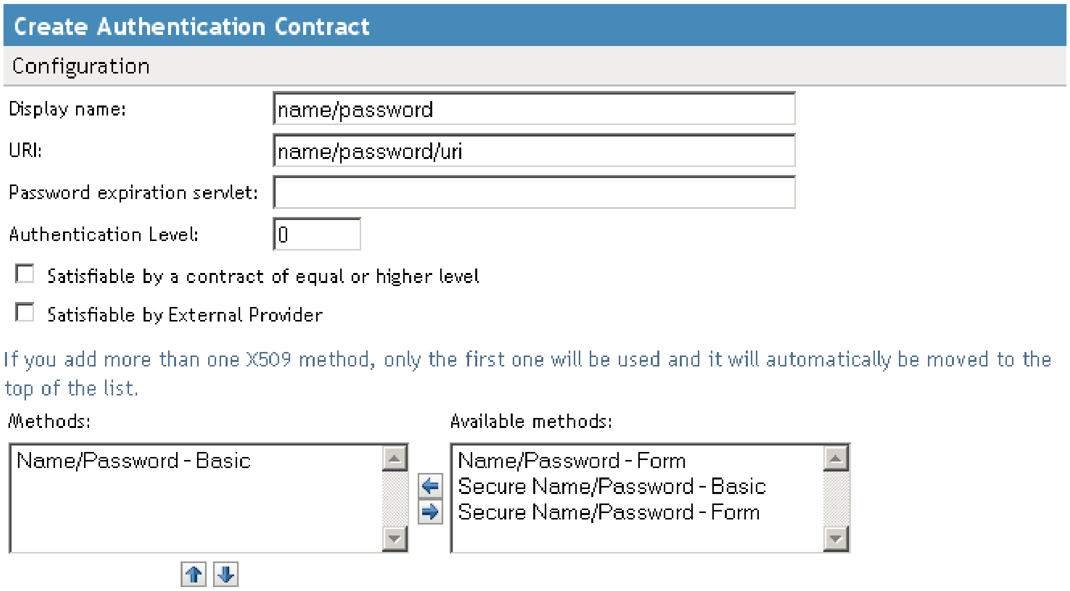
Authentication contracts define how authentication occurs. An Identity Server configuration might have several authentication contracts available, such as name/password or X.509, which is used for mutual SSL authentication between the Identity Server and the Access Gateway. Resources at an Access Gateway or agent are protected by authentication contracts.

NOTE: You cannot delete a contract if it is in use by an Access Gateway or J2EE agent.

Contracts are executed by the identity provider when authenticating a user. A URI uniquely identifies each contract, and you can assign authentication methods to each contract. A single contract can be specified for local logins.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Local* > *Contracts*

2 Click *New*.



Create Authentication Contract

Configuration

Display name: name/password

URI: name/password/uri

Password expiration servlet:

Authentication Level: 0

☐ Satisfiable by a contract of equal or higher level

☐ Satisfiable by External Provider

If you add more than one X509 method, only the first one will be used and it will automatically be moved to the top of the list.

Methods:
Name/Password - Basic

Available methods:
Name/Password - Form
Secure Name/Password - Basic
Secure Name/Password - Form

3 Fill in the following fields:

Display name: Specifies the name of the authentication contract.

URI: Specifies a value that uniquely identifies the contract from all other contracts. For example, as an identity provider, you might want to publish the details of a contract. In this case, you can use a URL so that the link resolves to a page. No spaces can exist in the URI field.

Password expiration servlet: Specifies a URL to a page where the user can change his or her password. This applies only to eDirectory servers when the password is expired or within the grace login period. You must use eDirectory to change the number of grace logins.

IMPORTANT: Failure to specify a value for the number of grace logins causes the contract to redirect to the password management servlet only when the grace login count has reached 0 and the password has expired.

The Identity Server needs to read the value of the grace login attribute in order to properly redirect to the password management servlet. If restricting grace logins is not important to your security model, enable grace logins and set the maximum to 9999 (the equivalent of infinite in most environments). For more information, see TID 3465171 (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3465171&sliceId=SAL_Public&dialogID=55170068&stateId=0%200%2055168646).

Access Manager works with any password management servlet, but for an implementation example, see Configuring Access Manager for UserApp and SAML (http://www.novell.com/coolsolutions/appnote/19981.html).

Authentication Level: A number you can assign to this authentication contract to specify its security level or rank. You use this setting to preserve authentication contracts of a higher security level. When you enable the *Satisfiable by a contract of equal or higher level* option on this page, the system uses this value as a reference.

For example, you might create a name/password authentication contract and assign it to level one. You might also create an X.509 authentication contract and assign it to level two. If a user supplies the credentials for the X.509 level-two contract, the system does not require the credentials to satisfy the name/password level-one authentication contract.

Satisfiable by a contract of equal or higher level: Allows the system to satisfy this authentication contract if a user has logged in using another contract of an equal or higher authentication level, as specified in the *Authentication Level* field of an authentication contract.

Satisfiable by External Provider: Enables your server to be authenticated by an external Identity Server. If your server and an external server are capable of performing an authentication, they must have the same URI.

Methods and Available Methods: Specifies the authentication method to use for the contract. You can specify the order in which the methods are executed for login; however, this is not a graded list, so all the methods you specify are required. *Available methods* are the authentication methods you have set up.

If you add more than one X.509 method, only the first one is used and it is automatically moved to the top of the list.

When choosing a secure method, such as Secure Name/Password, ensure that you have enabled security for the Identity Server configuration by setting the protocol to HTTPS. See "Configuring Secure Communication on the Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

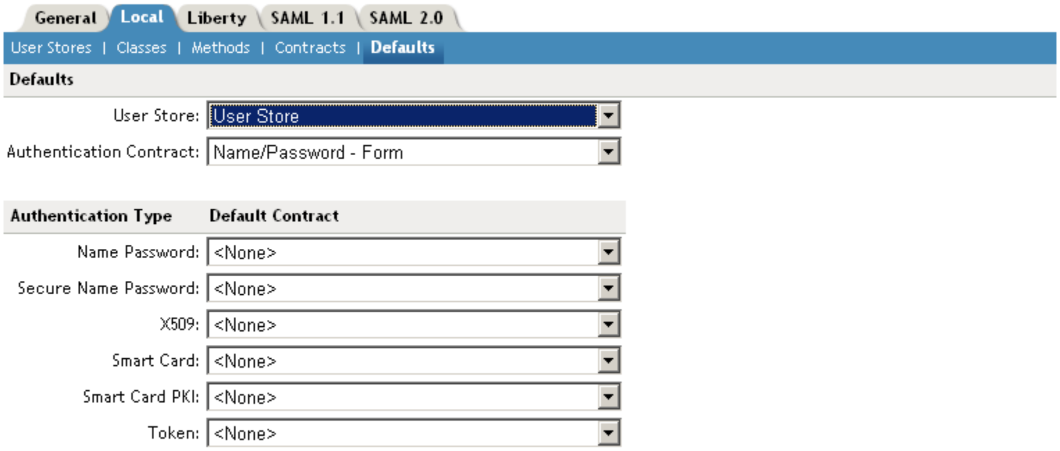
4. Click *Finish*, then *OK*.
5. Update the Identity Server and any devices that use the Identity Server configuration.

8.5 Specifying Authentication Defaults

You can specify default values for how the system processes user stores and authentication contracts. The default contract is executed when users access the system without a specified contract, and when the Access Gateway is configured to use any authentication.

Additional default contracts can be specified for each authentication type that might be required by a service provider. These contracts are executed when a request for a specific authentication type comes from a service provider.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Local* > *Defaults*



Defaults

User Store: User Store

Authentication Contract: Name/Password - Form

Authentication Type	Default Contract
Name Password:	<None>
Secure Name Password:	<None>
X509:	<None>
Smart Card:	<None>
Smart Card PKI:	<None>
Token:	<None>

2 Configure the following fields as necessary:

User Store: The default user store for local authentication. If you selected *<Default Userstore>* when configuring an authentication method, the system uses the user store you specify here.

Authentication Contract: The default authentication contract to be used for local authentication. If you create a new contract and specify it as the default one, ensure that you update the Access Gateway configuration if it is configured to accept the default (*Any*) contract. See Section 13.4, “Configuring Protected Resources,” on page 207.

Authentication Type: The default authentication contracts to be used for each authentication type. The identity provider uses the default authentication contract specified here, when the identity provider receives an authentication request from a service provider for a specific authentication type.

You must create the authentication contracts prior to assigning them as defaults. (See “Configuring Authentication Contracts” on page 111.)

3 Click *OK*.

4 Update the Identity Server.

8.6 Setting Up Mutual SSL Authentication

Mutual authentication is used when a user is issued a certificate from a trusted source. The certificate identifies the user in some way. To ensure the validity of X.509 certificates, Access

Manager supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) methods of verification.

SSL provides:

- Authentication and nonrepudiation of the server, using digital signatures
- Data confidentiality through the use of encryption
- Data integrity through the use of authentication codes

Mutual SSL provides the same things as SSL, with the addition of authentication and nonrepudiation of the client, using digital signatures.

1. Set up Access Manager certificates for security, and import them into the Access Manager system. (See Section 24.1, "Creating Certificates," on page 357.)
2. Create an X.509 authentication class. (Section 8.2.2, "Creating an X.509 Authentication Class," on page 104.)
3. Create an authentication method using this class. (Section 8.3, "Configuring Authentication Methods," on page 109.)
4. Create an authentication contract using the X.509 method. (Section 8.4, "Configuring Authentication Contracts," on page 111.)
5. Update any associated Access Gateways to read the new authentication contract. (Section 13.4, "Configuring Protected Resources," on page 207.)
6. Update the Identity Server cluster configuration. (See Section 3.2.1, "Updating an Identity Server Configuration," on page 38.)

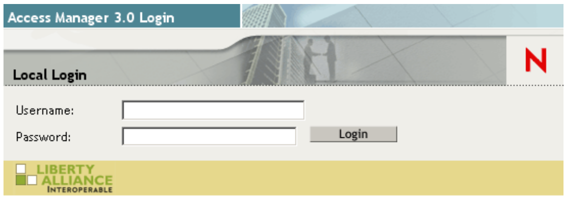
8.7 Creating Custom Login Pages

You can create custom login pages that refer to the Identity Server. You might want to rebrand the User Portal, authenticate users with non-default attributes (cn), or authenticate users based on multiple LDAP attributes. You also might be fronting several protected resources with an Access Gateway, and you need to create a unique login for each page.

- Section 8.7.1, "Modifying the Login Page," on page 114
- Section 8.7.2, "Creating Your Own Login or Error Page," on page 115

8.7.1 Modifying the Login Page

The following page is the default login page provided by the Access Manager. This page has been designed for the form-based authentication class.



Access Manager uses a JSP file as the default login page. You must be familiar with customizing `.jsp` files when creating custom login pages. The login page is located on the Identity Server in the following directory:

```
/var/opt/novell/tomcat4/webapps/nidp/jsp
```

You use the property name and values in the authentication classes and methods to customize the login. (See Section 8.2, “Creating Authentication Classes,” on page 101.).

The Radius and Protected classes also support a JSP property. You can use other classes, but if you want to create a custom login page, you must select a class that supports the JSP property. You can add this property to either the class or to the method derived from the class.

Property Names and Values

The default Property Name is Ecom_User_ID with a value of cn. You could, for example, change this to Ecom_User_ID and mail if you want to authenticate by using the user’s e-mail address. If you want to authenticate with the current username and password credentials, as well as the user’s e-mail address, you could modify the login page with an additional field in the form. For example:

input type="text" class="smalltext" name="Ecom_User_eMail" size="30"

You could then add an Ecom_User_eMail property name with “mail” as the property value. This is an example of an AND-based authentication request where you add the username AND user e-mail AND user password. You can OR fields if you add a Query property with a value similar to

(&(objectclass=person)(cn=%Ecom_User_ID%)(mail=%Ecom_User_eMail%)).

This entry allows you to add a field to the login page and allow the user to log in with a username or an e-mail address.

8.7.2 Creating Your Own Login or Error Page

The easiest way to create a new login page is to copy the default JSP page, rename it, and then modify it to match your requirements.

Login requirements:

- **Post Action:** https://IdentityServerDNS:8443/nidp/app/login
- **User name input type = “text”:** name=“Ecom_User_ID”
- **User password input type = “password:** name=“Ecom_Password”
- **Optional input type = “hidden”:** name=“target” with a value of a destination URL.

The default authentication contract is used if the post comes from an external page.

Logout links:

- **Identity Server:** https://IdentityServerDNS:8443/nidp/app/logout
- **Access Gateway:** https://AGAuthDomain/nesp/app/plogout

The location of the logout page for the Access Gateway:

- For the NetWare Access Gateway: sys:tomcat\4\webapps\nesp\jsp\
- For the Linux Access Gateway: /var/opt/novell/tomcat4/webapps/nesp/jsp/

To create a custom error page for the Identity Server, you must modify the `err.jsp` located in the following directory:

```
/var/opt/novell/tomcat4/webapps/nidp/jsp
```

To create custom error pages for the Access Gateway, see Section 15.5, "Customizing Error Pages," on page 256.

See Also

Customizing Error Messages in Access Manger Login Pages (http://www.novell.com/coolsolutions/feature/19987.html)

8.8 Managing Direct Access to the Identity Server

Users usually log into the Identity Server when they request access to a Web resource. They are redirected by the Access Gateway from the resource to the Identity Server to provide the required credentials for the resource. After they are authenticated, they are not prompted for credentials again, unless a resource requires credentials that they haven't already supplied.

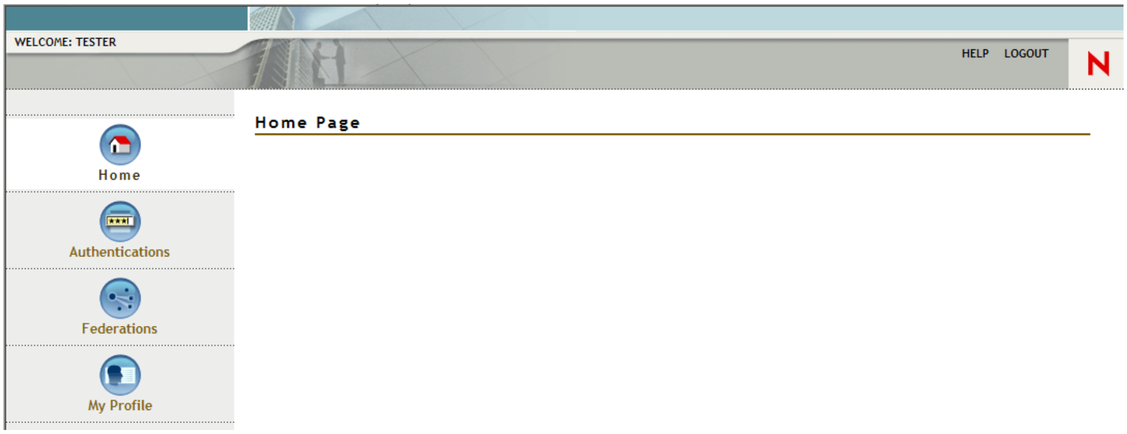
However, users can log directly into the Identity Server and access the User Portal, or they can access information about available Web Services Description Language (WSDL) services. This section describes how to manage access to these pages.

- Section 8.8.1, "Logging In to the User Portal," on page 116
- Section 8.8.2, "Blocking Access to the User Portal," on page 117
- Section 8.8.3, "Blocking Access to the WSDL Services Page," on page 117

8.8.1 Logging In to the User Portal

Users can log directly into the Identity Server when they enter the Base URL of the Identity Server in their browsers. For example, if your base URL is http://bfrei.provo.novell.com:8080/nidp, entering this URL prompts the user to authenticate with the credentials required for the default contract. If successful, the user is redirected to /nipd/app and the following page is displayed:

Figure 8-3 *User Portal*



This User Portal does not contain much useful information for the user unless you have set up federation or have enabled the Liberty profiles. (See Chapter 10, “Configuring User Authentication and Federation,” on page 157 and Chapter 12, “Configuring Liberty Web Services,” on page 173.)

8.8.2 Blocking Access to the User Portal

If you do not want users to have access to this User Portal page, you can disable direct login to the Identity Server by modifying a JSP page.

After a user successfully authenticates to the NIDP server directly, the `main.jsp` page from `/opt/novell/nids/lib/webapp/jsp` is presented to the user. This page builds the portal page with links to the `banner.jsp`, `nav.jsp`, `federations.jsp`, and `home.jsp`, which are in the same directory. The beginning lines of the `main.jsp` page build an HTTP response header. Find the following lines in the file:

```
<%
    response.setHeader("Pragma", "No-cache");
    response.setHeader("Cache-Control", "no-cache");
```

To avoid building the entire portal page that you do not want the users to access, inject an HTTP redirect so that users directly accessing the NIDP server are redirected to a page that you want them to access. For example to redirect users to novell.com, add the following line below the setHeader command:

```
    response.sendRedirect("http://www.novell.com");
```

Users are redirected to http://www.novell.com rather than to /nidp/app.

After saving the file, you do not need to restart Tomcat or the NIDP server. The changes should be effective immediately.

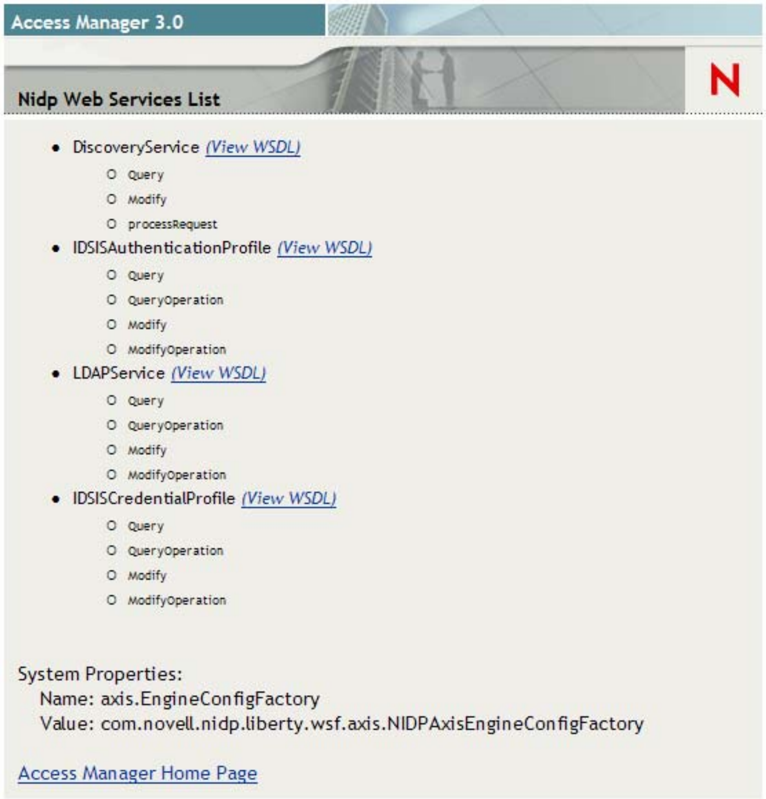
8.8.3 Blocking Access to the WSDL Services Page

Users can access the WSDL service page when they enter the Base URL of the Identity Server in their browsers with the path to the Services page. For example, if your base URL is http://bfrei.provo.novell.com:8080/nidp, the users can access the services page with the following URL:

```
http://bfrei.provo.novell.com:8080/nidp/services
```

The Services page contains the following information and links:

Figure 8-4 WSDL Services Page



Access Manager 3.0

Nidp Web Services List

- DiscoveryService (View WSDL)
    - Query
    - Modify
    - processRequest
- IDSISAuthenticationProfile (View WSDL)
    - Query
    - QueryOperation
    - Modify
    - ModifyOperation
- LDAPService (View WSDL)
    - Query
    - QueryOperation
    - Modify
    - ModifyOperation
- IDSISCredentialProfile (View WSDL)
    - Query
    - QueryOperation
    - Modify
    - ModifyOperation

System Properties:
Name: axis.EngineConfigFactory
Value: com.novell.nidp.liberty.wsf.axis.NIDPAxisEngineConfigFactory

Access Manager Home Page

If you do not want your users to have access to this page, you can block access by modifying the `web.xml` file in the `/opt/novell/nids/lib/webapp/WEB-INF` directory. Near the top of the file, in the context initialization parameters section, add the following lines:

```
<context-param>
      <param-name>wsfServicesList</param-name>
      <param-value>full</param-value>
</context-param>
```

When `<param-value>` has a value of `full`, users can access the Services page. To modify this behavior, replace `full` with one of the following values:

Table 8-1 *Context Parameter Values*

Value	Description
404	Returns an HTTP 404 status code: Not Found
403	Returns an HTTP 403 status code: Forbidden
empty	Returns an empty services list

If the parameter is removed from the file or if you enter an invalid value, the value is interpreted as `full`, and users have access to the page.

You need to restart Tomcat for your modifications to take effect:

```
/etc/init.d/novell-tomcat4 restart
```

8.9 Configuring Kerberos for Authentication

Kerberos is an authentication method that allows users to log in to an Active Directory domain. This authentication method provides them with a token, which an Identity Server can be configured to use as a contract. This provides single sign-on for the user between Active Directory and the Identity Server.

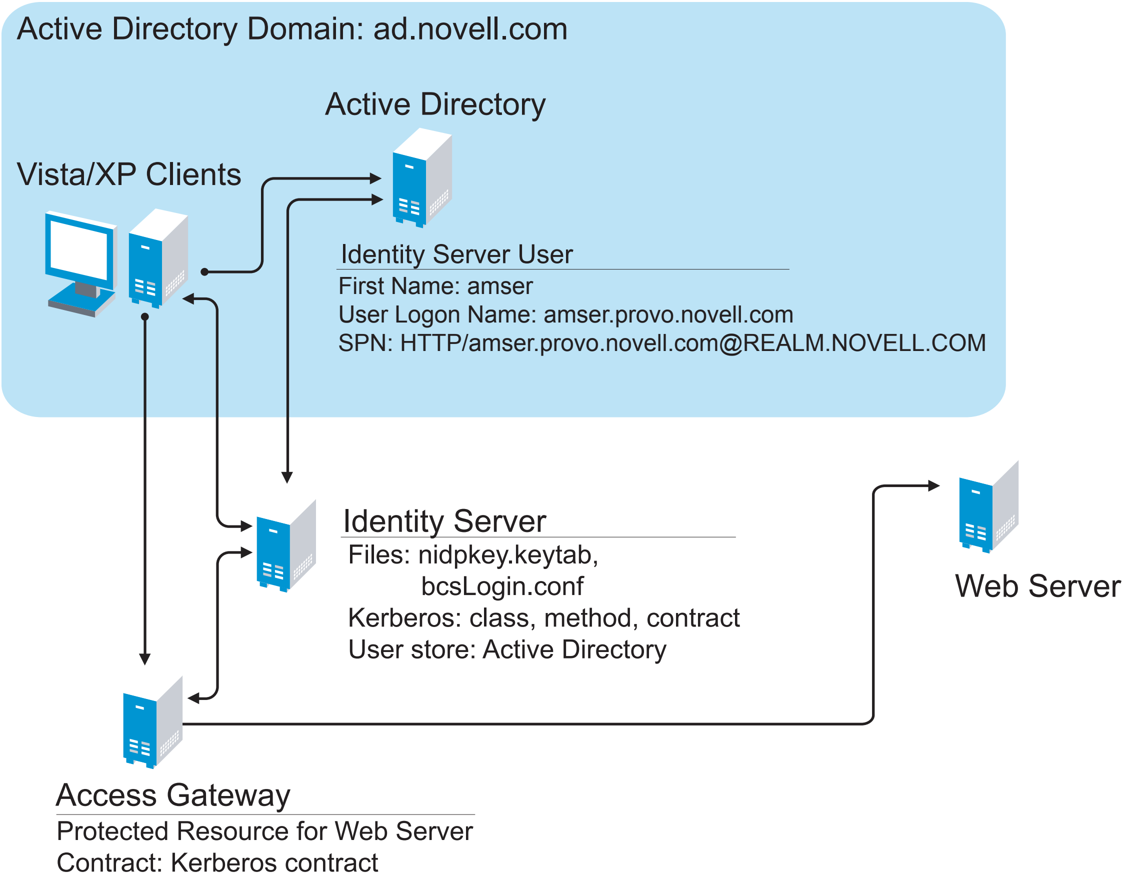
Kerberos authentication is achieved using SPNEGO with GSS-API (JGSS). SPNEGO (RFC 2478 - Simple and Protected GSSAPI Negotiation implementation in Microsoft* Windows 2000/XP/2k3) is a GSSAPI mechanism for extending a Kerberos based single-sign-on environment to Web transactions and services. It lets peers determine which GSSAPI mechanisms are shared and lets them select one and establish a security context with it. SPNEGO's most visible use is in Microsoft's HTTP Negotiate authentication mechanism.

The Kerberos module for Access Manager is implemented as additional out-of-the-box authentication mechanism to securely negotiate and authenticate HTTP requests for protected resources. This makes it possible to seamlessly authenticate (single-sign-on) to the Identity Server from enterprise-wide Microsoft Windows Domain Logon.

In situations where the system cannot use the Kerberos configuration, such as if the browser is trying to authenticate from outside of a firewall and fails, the fallback authentication methods are NTLM (which Access Manager does not use), then HTTPS basic authentication. This can cause the system to prompt users twice for authentication. (To disable this in Windows Explorer, click *Tools* > *Internet Options* > *Security* > *Custom Level*, then scroll down to *User Authentication*. Enable *Automatic logon with current user name and password*.)

This section explains how to configure Active Directory, the Identity Server, and the Access Gateway for Kerberos authentication to a protected Web server. Figure 8-5 illustrates this configuration.

Figure 8-5 Example Kerberos Configuration


Active Directory Domain: ad.novell.com
Active Directory
Vista/XP Clients
Identity Server User
First Name: amser
User Logon Name: amser.provo.novell.com
SPN: HTTP/amser.provo.novell.com@REALM.NOVELL.COM
Identity Server
Files: nidpkey.keytab,
bcsLogin.conf
Kerberos: class, method, contract
User store: Active Directory
Web Server
Access Gateway
Protected Resource for Web Server
Contract: Kerberos contract

Kerberos requires the following configuration tasks:

- Section 8.9.1, "Prerequisites," on page 120
- Section 8.9.2, "Configuring Active Directory," on page 121
- Section 8.9.3, "Configuring the Identity Server," on page 123
- Section 8.9.4, "Configuring the Clients," on page 128
- Section 8.9.5, "Configuring the Access Gateway for Kerberos Authentication," on page 129

8.9.1 Prerequisites

Kerberos authentication is supported for the following configuration:

- Clients must be running Windows XP with Internet Explorer 7. Some minimal testing has been done with Internet Explorer 6. The Windows Vista* client has had only cursory testing. If you have problems with the Vista client, please report these problems to Novell.
- Active Directory must be configured to contain entries for both the users and their machines. The Kerberos configuration was tested with Active Directory running on Windows 2003 Enterprise Server SP2. The configuration has not been tested with Active Directory running Windows 2000 Server.

- Your Access Manager components must be running Access Manager 3.0 SP2. The configuration was tested using the Linux Access Gateway. The NetWare Access Gateway was not used in the configuration, and therefore not tested for Kerberos compatibility.
- Active Directory and the Identity Server must be configured to use a Network Time Protocol server. If time is not synchronized, authentication fails.
- Both TCP and UDP need to be enabled. Kerberos defaults to TCP but only after failing on UDP (for example with packet-size limitations). This is a limitation with the underlying JGSS layer which cannot use TCP as the only transport protocol.

8.9.2 Configuring Active Directory

You must create a new user in Active Directory for the Identity Server, set up this user account to be a service principal, create a keytab file, and add the Identity Server to the Forward Lookup Zone. These tasks are described in the following sections:

- "Installing the spn and the ktpass Utilities" on page 121
- "Creating and Configuring the User Account for the Identity Server" on page 121
- "Configuring the Keytab File" on page 122
- "Adding the Identity Server to the Forward Lookup Zone" on page 123

Installing the spn and the ktpass Utilities

When you install Windows 2003 and Active Directory, the spn and ktpass utilities are not installed in a default installation. You need both of these utilities to configure the Identity Server for Kerberos authentication.

1. Insert the Windows 2003 CD into the CD drive.
2. To install the utilities, run `\SUPPORT\TOOLS\SUPTOOLS.MSI` on the CD.

   The utilities are installed in `C:\Program Files\Support Tools`.

Creating and Configuring the User Account for the Identity Server

1. In *Manage Your Server* on your Windows 2003 server, select the *Manage users and computers in Active Directory* option.
2. Select to create a new user.
3. Fill in the following fields:

   First name: Specify the hostname of the Identity Server. This is the username. For the example configuration, this is `amser`.

   User logon name: Specify `HTTP/<Identity_Server_DNS_name>`. For this example configuration, your Identity Server has a hostname of `amser` and a domain name of `provo.novell.com`. For these names, you would specify the following for the *User Logon Name*:

   `HTTP/amser.provo.novell.com`

   The realm is displayed next to the *User logon name*.

   User logon name (pre Windows 2000): Specify the hostname of the Identity Server. The default value must be modified. For the example configuration, this is `amser`.
4. Click *Next*, and configure the password and its options:

Password: Specify a password for this user

Confirm password: Enter the same password.

User must change password at next logon: Deselect this option.

Password never expires: Select this option.

5 Click *Next*, then *Finish*.

This creates the Identity Server user. You need to remember the values you assigned to this user for *First name* and *User logon name*.

6 Right-click the user you just created, then select *Properties*.

7 Click the *Account* tab, then in *Account Options*, select *Use DES encryption types for this account*.

Section , "Configuring the Keytab File," on page 122 explains how to create a keytab file. If you ever change the encryption type for the Identity Server user, you need to reset the password for the user and regenerate the keytab file.

8 Click *Apply*, then *OK*.

9 Change the password of the user.

After setting up the account to use DES encryption, the password needs to be reset.

10 To set the servicePrincipalName (spn) attribute on this user, open a command window and enter the following command:

```
setspn -A HTTP/<userLogonName> <userName>
```

For this configuration example, you would enter the following command:

```
setspn -A HTTP/amser.provo.novell.com@REALM.NOVELL.COM amser
```

This adds the servicePrincipalName attribute to the user specified with the value specified in the `-A` parameter.

11 (Optional) Verify that the user has the required servicePrincipalName attribute with a valid value. Enter the following command:

```
setspn -L <userName>
```

For this configuration example, you would enter the following command:

```
setspn -L amser
```

Configuring the Keytab File

The keytab file contains the secret encryption key that is used to decrypt the Kerberos ticket. You need to generate the keytab file and copy it to the Identity Server.

1 On the Active Directory server, open a command window and enter a `ktpass` command with the following parameters:

```
ktpass /out value /princ value /mapuser value /pass value /crypto
value /DesOnly
```

The command parameters require the following values:

Parameter	Value	Description
/out	<outputFilename>	Specify a name for the file, with.keytab as the extension. For example: `nidpkey.keytab`

Parameter	Value	Description
/princ	<servicePrincipalName>@<KERBEROS_REALM>	Specify the service principal name for the Identity Server, then @, followed by Kerberos realm. The default value for the Kerberos realm is the Active Directory domain name in all capitals. The Kerberos realm value is case sensitive.
/mapuser	<identityServerUser>@<AD_DOMAIN>	Specify the username of the Identity Server user and the Active Directory domain to which the user belongs.
/pass	<userPassword>	Specify the password for this user.
/crypto	DES-CBC-MD5	Specify DES-CBC-MD5. This is the only cryptology algorithm supported in this release.
/DesOnly	N/A	This parameter enforces the use of DES cryptology, and does not require a value.

For this configuration example, you would enter the following command to create a keytab file named `nidpkey`:

```
ktpass /out nidpkey.keytab /princ HTTP/
amser.provo.novell.com@AD.NOVELL.COM
/mapuser amser@AD.NOVELL.COM /pass novell /crypto DES-CBC-MD5 /
DesOnly
```

2. Copy the keytab file to the Identity Server.

   Copy the file to the default location on the Identity Server:

   SLES 10: `/opt/novell/java/jre/lib/security`

   SLES 9: `/usr/lib/java/jre/lib/security`

Adding the Identity Server to the Forward Lookup Zone

1. In Manage Your Server on your Windows 2003 server, click *Manage this DNS server*.
2. Click *Forward Lookup Zone*.
3. Click the Active Directory domain.
4. In the right pane, right click, and select *New Host (A)*.
5. Fill in the following fields:

   Name: Specify the hostname of the Identity Server.

   IP Address: Specify the IP address of the Identity Server.
6. Click *Add Host*.

8.9.3 Configuring the Identity Server

You need to configure the Identity Server to use the Active Directory server as a user store, configure a Kerberos authentication class, method, and contract, create a configuration file, enable logging to verify the configuration, then restart tomcat. These instructions assume that you have installed and configured an Identity Server cluster configuration. If you have not, see the Novell Access Manager 3.0 Installation Guide (http://www.novell.com/documentation/

novellaccessmanager/installation/data/bookinfo.html) and the Novell Access Manager 3.0 Setup Guide (http://www.novell.com/documentation/novellaccessmanager/basicconfig/data/bookinfo.html).

This section covers the following tasks:

- "Configuring the Identity Server for Active Directory" on page 124
- "Creating the Authentication Class, Method, and Contract" on page 125
- "Creating the bcsLogin Configuration File" on page 127
- "Verifying the Kerberos Configuration" on page 127

Configuring the Identity Server for Active Directory

You need to either configure your Identity Server to use Active Directory as a user store or verify your existing configuration for your Active Directory user store.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.
2. Click *Local*.
3. View your installed user stores.

   If you have already configured your Identity Server to use the Active Directory server, click its name.

   If you haven't configured a user store for the Active Directory server, click *New*.
4. For a new user store, fill in the following fields. For an existing Active Directory user store, verify the values.

   Name: Specify a name of the user store for reference.

   Admin name: Specify the name of the administrator of the Active Directory server. Administrator-level rights are required for setting up a user store. This ensures read/write access to all objects used by Access Manager.

   Admin password and Confirm password: Specify the password for the administrator of the Active Directory server and confirm the password.

   Directory Type: Select *ActiveDirectory*.

   Search Contexts: For a new user store, click *New* and specify the context of the administrator of the Active Directory server. For an existing user store, verify that you have an entry for the context of the administrator and add one if it is missing.
5. (Conditional) For a new Active Directory user store, add a replica. In the *Server replicas* section, click *New*.

   5a Fill in the following fields:

   Name: Specify a name of the replica for reference. This can be the name of your Active Directory server.

   IP Address: Specify the IP address of the Active Directory server and the port you want the Identity Server to use when communication with the Active Directory server.

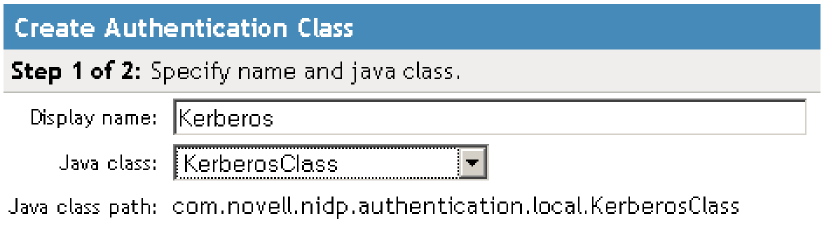
   5b Configure the other fields to fit your security model.

   5c Click *OK*.
6. (Optional) Specify values for the other configuration options.
7. To save your changes, click *OK* or *Finish*.

8 Continue with Section , "Creating the Authentication Class, Method, and Contract," on page 125.

Creating the Authentication Class, Method, and Contract

1 In the Local page, click *Classes* > *New*.



Create Authentication Class

Step 1 of 2: Specify name and java class.

Display name: Kerberos

Java class: KerberosClass

Java class path: com.novell.nidp.authentication.local.KerberosClass

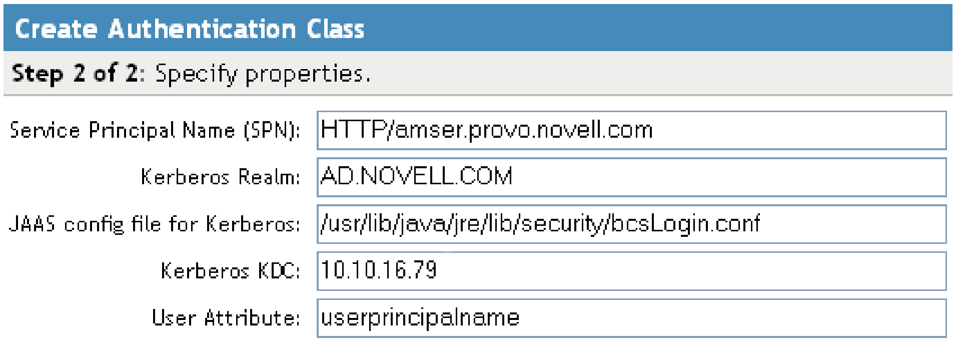
2 Fill in the following fields:

Display name: Specify a name that you can use to identify this class.

Java class: Select *KerberosClass*.

The *Java class path* field displays the name of the KerberosClass.

3 Click *Next.*



Create Authentication Class

Step 2 of 2: Specify properties.

Service Principal Name (SPN): HTTP/amser.provo.novell.com

Kerberos Realm: AD.NOVELL.COM

JAAS config file for Kerberos: /usr/lib/java/jre/lib/security/bcsLogin.conf

Kerberos KDC: 10.10.16.79

User Attribute: userprincipalname

4 Fill in the following fields:

Service Principal Name (SPN): Specify the value of the servicePrincipalName attribute of the Identity Server user. For this example configuration, this is `HTTP/amser.provo.novell.com`.

Kerberos Realm: Specify the name of the Kerberos realm. The default value for this realm is the domain name of the Active Directory server, entered in all capitals. The value in this field is case sensitive. For this example configuration, this is `AD.NOVELL.COM`.

JAAS config file for Kerberos: Verify the default path. This should be the same path to which you copied the keytab file (see Step 2 in Section , "Configuring the Keytab File," on page 122) and end with the name of the configuration file, `bcsLogin.conf`.

If you have not created this configuration file, see Section , "Creating the bcsLogin Configuration File," on page 127.

Kerberos KDC: Specify the IP address of the Active Directory server.

User Attribute: Specify the name of the Active Directory attribute that combines the cn of the user with the DNS domain name to form its value. It is an alternate name for user login. Accept the default value unless you have set up a different attribute.

5. Click *Finish*.
6. In the Local page, click *Methods* > *New.*
7. Fill in the following fields:

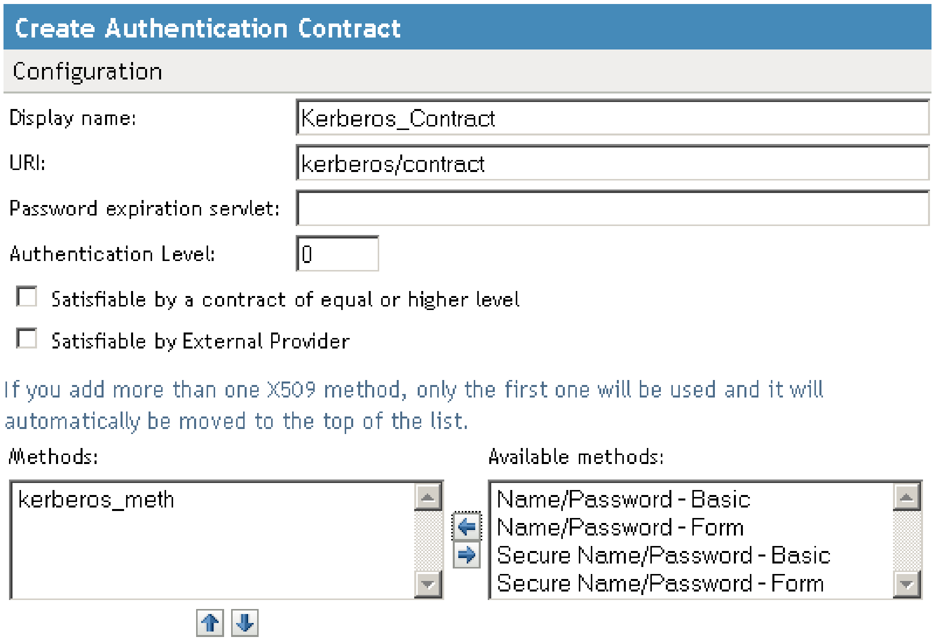
   Display name: Specify a name that you can use to identify this method.

   Class: Select the class that you created for Kerberos.

   User stores: Move the Active Directory user store to the list of User stores. If you have only one installed user store, <Default User Store> can be used. If you have multiple user stores, the Active Directory user store must be in this list (or if it is configured to be the default user store, <Default User Store> must be in this list).

   NOTE: The testing procedure to verify Kerberos authentication is dependent upon having the Active Directory user store configured as the default user store. See Step 12.

   You do not need to configure properties for this method.
8. Click *Finish*.
9. In the Local page, click *Contracts* > *New.*



Create Authentication Contract

Configuration

Display name: Kerberos_Contract

URI: kerberos/contract

Password expiration servlet:

Authentication Level: 0

☐ Satisfiable by a contract of equal or higher level

☐ Satisfiable by External Provider

If you add more than one X509 method, only the first one will be used and it will automatically be moved to the top of the list.

Methods:
kerberos_meth

Available methods:
Name/Password - Basic
Name/Password - Form
Secure Name/Password - Basic
Secure Name/Password - Form

10. Fill in the following fields:

    Display name: Specify a name that you can use to identify this method.

    URI: Specify a value that uniquely identifies the contract from all other contracts.

    The URI cannot begin with a slash, and it must uniquely identity the contract. For example: `kerberos/contract`

    Methods: From the list of *Available methods*, move your Kerberos method to the *Methods* list.

    You do not need to configure the other contract options.

11 Click *Finish*.

12 (Optional) To use the procedure that verifies the authentication configuration, you need to make the Active Directory user store the default user store. In the Local page, click *Defaults*.

12a Fill in the following fields:

User Store: Select the name of your Active Directory user store.

Authentication Contract: Select the name of your Kerberos contract.

12b Click *OK*.

This allows you to log in directly to the Identity Server using the Kerberos contract. If you have already logged in to the Active Directory domain on the Windows machine, single sign-on is enabled and you are not prompted to log in to the Identity Server.

13 On the Identity Servers page, click *Update*.

Wait until the Health icon turns green. Click *Refresh* to update the page.

14 If you have Access Gateways or J2EE Agents that you want to configure to use the Kerberos contract, update these devices so that the Kerberos contract is available.

15 Continue with Section , "Creating the bcsLogin Configuration File," on page 127.

Creating the bcsLogin Configuration File

1 Open a text editor.

2 Enter the following lines. The file cannot contain any white space, only end of line characters. Two lines (principal and keyTab) need to specify unique information for your configuration. The principal line needs to specify the service principle name for the Identity Server. The keyTab line needs to specify the location of the keytab file. The following file uses the values of the example configuration for the principal and keyTab lines. The keyTab and ticketCache lines use the default path for SLES 10. For SLES 9, you need to use `/usr/lib/java/jre/lib/security` for the path.

```
other {
com.sun.security.auth.module.Krb5LoginModule required
debug="true"
useTicketCache="true"
ticketCache="/opt/novell/java/jre/lib/security/spnegoTicket.cache"
doNotPrompt="true"
principal="HTTP/amser.provo.novell.com@AD.NOVELL.COM"
useKeyTab="true"
keyTab="/opt/novell/java/jre/lib/security/nidpkey.keytab"
storeKey="true";
};
```

3 Save this file with a name of `bcsLogin.conf`.

4 Copy this file to the location specified in the *JAAS config file for Kerberos* field of Step 4 in Section , "Creating the Authentication Class, Method, and Contract," on page 125.

5 Restart Tomcat. In a command window on the Identity Server, enter the following command.

```
/etc/init.d/novell-tomcat4 restart
```

Whenever you make changes to the `bcsLogin.conf` file, you need to restart Tomcat.

Verifying the Kerberos Configuration

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Logging*.

2 Enable the *File Logging* and *Echo To Console* options.

3 In the *Component File Logger Levels* section, set *Application* to *debug*.

4 In the *Trace Logging* section, select *Enabled*.

5 Select *Application* and *Configuration* as *Component Content Filters*:

6 Click *OK*, then *Update*.

7 View the `catalina.out` file of the Identity Server. In the Administration Console, click *Access Manager* > *Auditin*g.

  7a Click *General Logging*.

  7b In the Identity Servers section, click the link to the `catalina.out` file.

  7c Open the file in a text editor.

  7d Search for Kerberos and verify that a subsequent line contains `Commit Succeeded`. For the configuration example, the lines look similar to the following:

```
principal's key obtained from the keytab
principal is HTTP/amser.provo.novell.com@AD.NOVELL.COM
Added server's keyKerberos Principal HTTP/
amser.provo.novell.com@AD.NOVELL.COMKey Version 3key
EncryptionKey: keyType=3 keyBytes (hex dump)=0000: CB 0E 91 FB
7A 4C 64 FE

[Krb5LoginModule] added Krb5Principal HTTP/
amser.provo.novell.com@AD.NOVELL.COM to Subject
Commit Succeeded
```

  7e If the commit did not succeed, search backwards in the file and verify the following values:

  - Service Principal Name
  - Name of keytab file

  For the example configuration, the file would contains lines with text similar to the following:

```
Principal is HTTP/amser.provo.novell.com

KeyTab is /usr/lib/java/jre/lib/security/nidpkey.keytab
```

  7f If you make any modifications to the configuration, either in the Administration Console or to the bcsLogin file, restart Tomcat on the Identity Server.

8.9.4 Configuring the Clients

1 Add the computers of the users to the Active Directory domain.

For instructions, see your Active Directory documentation.

2 Log in to the Active Directory domain, rather than the machine.

3 Configure the Web browser to trust the Identity Server:

- For Internet Explorer version 7, click *Tools* > *Internet Options* > *Security* > *Local intranet* > *Sites* > *Advanced*. (For Internet Explorer version 6, click *Tools* > *Internet Options* > *Security* > *Trusted sites* > *Sites*.)

In the *Add this website to the zone* text box, enter the Base URL for the Identity Server, then click *Add*.

In the configuration example, this is `http://amser.provo.novell.com`.

Click *Close*.

- For Firefox, in the URL field, specify `about:config`. In the *Filter* field, specify *network.n*. Double click `network.negotiate-auth.trusted-uris`.

  This preference lists the sites that are permitted to engage in SPNEGO Authentication with the browser. Enter a comma-delimited list of trusted domains or URLs.

  For this example configuration, you would add `http://amser.provo.novell.com` to the list.

  If the deployed SPNEGO solution is using the advanced Kerberos feature of Credential Delegation, double click `network.negotiate-auth.delegation-uris`. This preference lists the sites for which the browser can delegate user authorization to the server. Enter a comma-delimited list of trusted domains or URLs.

  For this example configuration, you would add `http://amser.provo.novell.com` to the list.

4. Click *OK*. The configuration appears as updated.

   Restart your Firefox browser to activate this configuration.

5. In the URL field, enter the base URL of the Identity Server with port and application. For this example configuration:

   `http://amser.provo.novell.com:8080/nidp`

   The Identity Server should authenticate the user without prompting the user for authentication information. If a problem occurs, check for the following configuration errors:

   - Verify the default user store and contract. See Step 12.
   - View the `catalina.out` file and verify the configuration. See “Verifying the Kerberos Configuration” on page 127.
   - If you make any modifications to the configuration, either in the Administration Console or to the `bcsLogin` file, restart Tomcat on the Identity Server.
   - To verify whether a Kerberos ticket is being sent to the client, download the Kerbtray Tool (http://www.microsoft.com/downloads/details.aspx?familyid=4E3A58BE-29F6-49F6-85BE-E866AF8E7A88&displaylang=en) and run it on the client workstation.
   - Enable Kerberos event logging. See How to enable Kerberos event logging (http://support.microsoft.com/kb/262177/).
   - Make sure all components have their time synchronized.

8.9.5 Configuring the Access Gateway for Kerberos Authentication

If you have set up a Web server that you want to require Kerberos authentication for access, you can set up a protected resource for this Web server as you would for any other Web server, and select the name of your Kerberos contract for the contract. For instructions, see Section 13.4, “Configuring Protected Resources,” on page 207.

8.10 Configuring Access Manager for NESCM

To use a smart card with Access Manager, you need to configure Access Manager to use the eDirectory server where you have installed the NESCM method. You then need to create a contract that knows how to prompt the user for the smart card credentials. The last task is to assign this contract to the protected resources that you want protected with a smart card. The following sections describe prerequisites and the tasks:

- Section 8.10.1, "Prerequisites," on page 130
- Section 8.10.2, "Creating a User Store," on page 130
- Section 8.10.3, "Creating a Contract for the Smart Card," on page 132
- Section 8.10.4, "Assigning the NESCM Contract to a Protected Resource," on page 135
- Section 8.10.5, "Verifying the User's Experience," on page 136
- Section 8.10.6, "Troubleshooting," on page 137

8.10.1 Prerequisites

- ☐ Make sure you can authenticate to the eDirectory server using the smart card from a workstation.
    - The NESCM method needs to be installed on the eDirectory server and the workstation. See Installing the Method (http://www.novell.com/documentation/ias303/nescm_install/data/b7gx5la.html).
    - The NESCM method needs to be configured. See Basic Configuration Requirements (http://www.novell.com/documentation/ias303/nescm_install/data/b7tf2gi.html).
    - Provision your smart card according to your company policy.
- ☐ Make sure you have a basic Access Gateway configuration with a protected resource that you want to protect with a smart card. For more information, see the *Novell Access Manager 3.0 SP3 IR2 Installation Guide* and the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

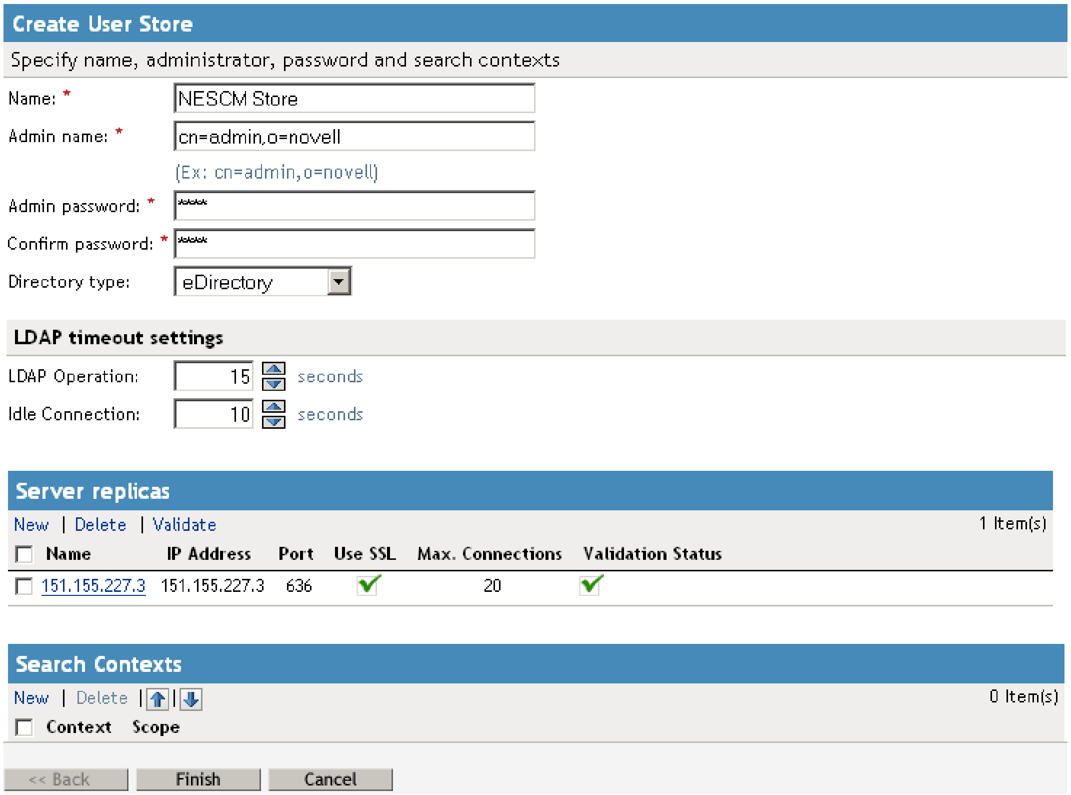
8.10.2 Creating a User Store

The Identity Server must be configured to use the eDirectory replica where you have installed the NESCM server method.

- If you have already configured the Identity Server to use this replica, skip this section and continue with Section 8.10.3, "Creating a Contract for the Smart Card," on page 132.
- If your Identity Server is using a different user store, you need to configure the Identity Server.

To configure the Identity Server for the eDirectory replica that has the NESCM method:

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Local*> *User Stores* > *New*.



Create User Store

Specify name, administrator, password and search contexts

Name: * NESCM Store

Admin name: * cn=admin,o=novell

(Ex: cn=admin,o=novell)

Admin password: * ****

Confirm password: * ****

Directory type: eDirectory

LDAP timeout settings

LDAP Operation: 15 seconds

Idle Connection: 10 seconds

Server replicas

New | Delete | Validate 1 Item(s)

☐ Name	IP Address	Port	Use SSL	Max. Connections	Validation Status
☐ 151.155.227.3	151.155.227.3	636	✓	20	✓

Search Contexts

New | Delete 0 Item(s)

☐ Context	Scope

2. On the *Create User Store* page, fill the following fields:

   Name: A display name for the eDirectory replica (for example, `nescm_replica`).

   Admin Name: The distinguished name of the admin user of the directory. Administrator-level rights are required for setting up a user store.

   Admin Password and Confirm Password: The password for the admin user and the confirmation for the password.

   Directory Type: Select eDirectory.

3. In the Server replica section, click *New*, and fill the following fields:

   Name: The display name for the LDAP directory server (for example, `nescm_server`).

   IP Address: The IP address of the LDAP directory server. The port is set automatically to the standard LDAP ports.

4. Click *Use secure LDAP connections*. You must enable SSL between the user store and the Identity Server. The port changes to 636, the secure LDAP port.

5. Click *Auto import trusted root*.

6. Click *OK* to confirm the import.

7. Select the *Root CA Certificate* to trust any certificate signed by that certificate authority.

8. Specify an alias, then click *OK*.

   An alias is a name you use to identify the certificate used by Access Manager.

9. Click *Close*, then click *OK*.

10. Under *Server Replicas*, verify the *Validation Status*.

    The system displays a green check mark if the connection is valid.

11 (Optional) Set up a search context.

12 Click *Finish* to save the information.

13 Continue with Section 8.10.3, “Creating a Contract for the Smart Card,” on page 132

8.10.3 Creating a Contract for the Smart Card

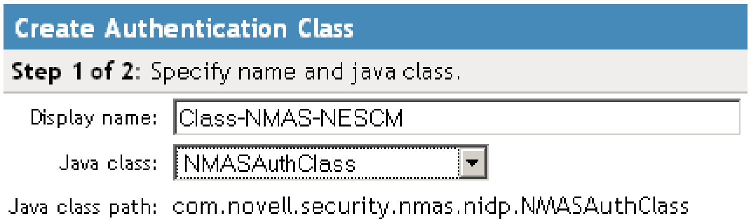
You need to create a contract that uses the NESCM method. To do this, you need to first create an NMAS class, then a method that uses that class. The last task is to create a contract that uses the method. The following sections describe these tasks:

- “Creating an NMAS Class for NESCM” on page 132
- “Creating a Method to Use the NMAS Class” on page 133
- “Creating an Authentication Contract to Use the Method” on page 134

Creating an NMAS Class for NESCM

When you create a class, you can specify values for properties. In the following steps, you specify a property value that determines the sequence of login prompts that the user receives when authenticating with a smart card.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Local* > *Classes* > *New*.



Create Authentication Class

Step 1 of 2: Specify name and java class.

Display name: Class-NMAS-NESCM

Java class: NMASAuthClass

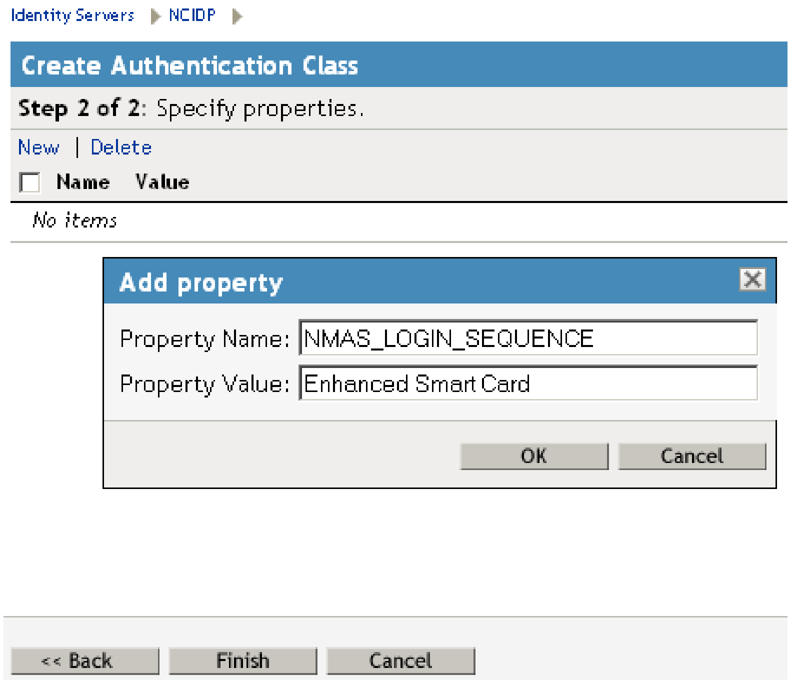
Java class path: com.novell.security.nmas.nidp.NMASAuthClass

2 Specify a name for the class *Display name* (for example, `Class-NMAS-NESCM`).

3 For the *Java class*, select *NMASAuthClass* from the selection list.

4 Click *Next*.

5 On the *Specify Properties* page, click *New*.



Step 2 of 2: Specify properties.

No items

Property Name: NMAS_LOGIN_SEQUENCE

Property Value: Enhanced Smart Card

6 Specify the following values for the property:

 Property Name: Specify `NMAS_LOGIN_SEQUENCE`

 Property Value: Specify `Enhanced Smart Card`

 These values match the method name as displayed in *NMAS* task > *NMAS Login Methods*.

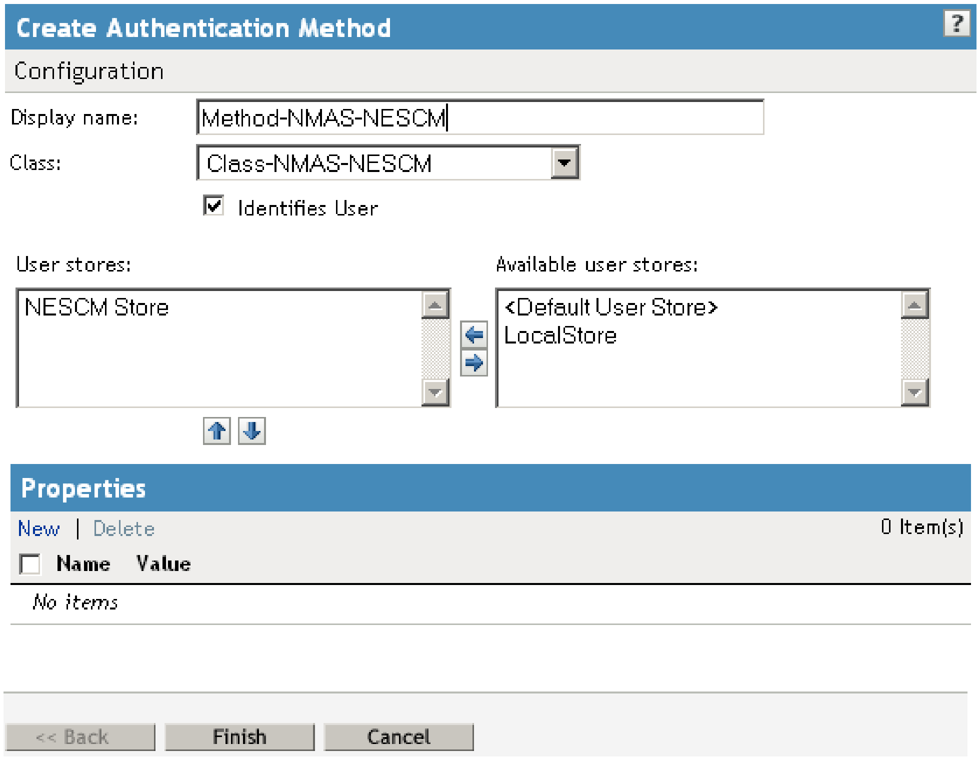
7 Click *OK*, then *Finish*.

8 Continue with "Creating a Method to Use the NMAS Class" on page 133

Creating a Method to Use the NMAS Class

When creating a method, you can specify property values that are applied to just this method and not the entire class. In this tutorial, we want the method to use the same login sequence as the class. The method also allows you to specify which user stores can use the method. For a smart card method, you need to ensure that the user store or stores specified for the method have NESCM installed.

1 On the Local page for the Identity Server, click *Methods* > *New*.



Display name: Method-NMAS-NESCM

Class: Class-NMAS-NESCM

☑ Identifies User

User stores: NESCM Store

Available user stores: <Default User Store> LocalStore

No items

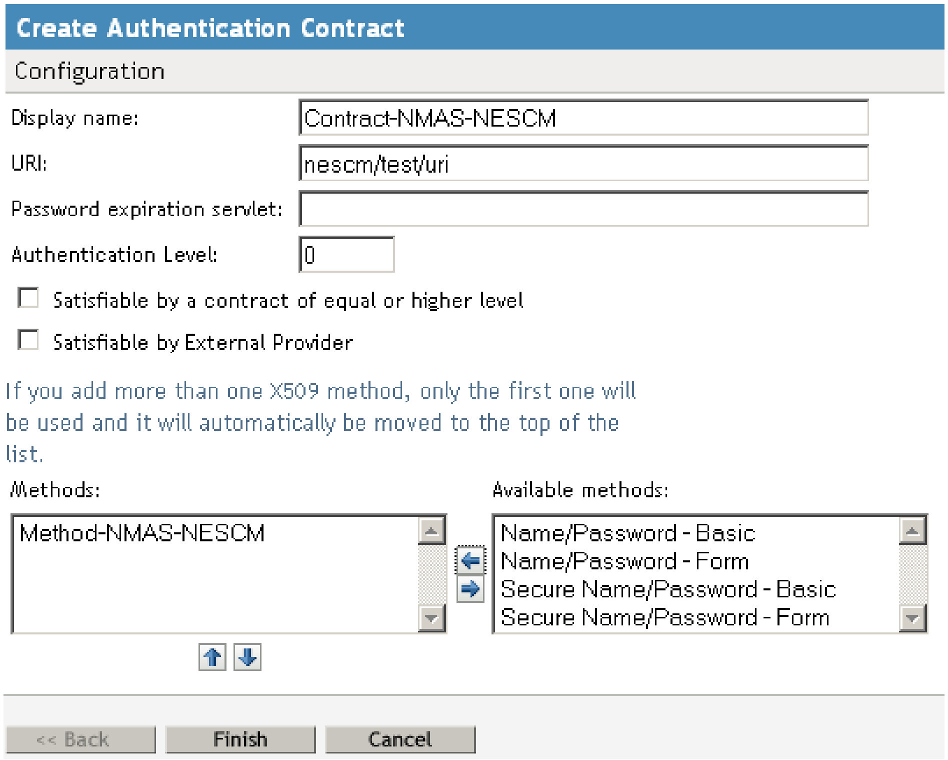
2. Specify a *Display name* (for example, `Method-NMAS-NESCM`).
3. From the *Class* selection list, select the class created in "Creating an NMAS Class for NESCM" on page 132.
4. In the *Available user stores list*, select the user store created in Section 8.10.2, "Creating a User Store," on page 130, then click the left-arrow to move this user store into the *User stores* list.

   Leave other settings on this page unchanged.
5. Click *Finish*.
6. Continue with "Creating an Authentication Contract to Use the Method" on page 134.

Creating an Authentication Contract to Use the Method

Contracts are the element you can assign to a protect a resource. Because NESCM uses certificates, you should assign only one method to a contract.

1. On the Local page for the Identity Server, click *Contracts* > *New*.



Create Authentication Contract

Configuration

Display name: Contract-NMAS-NESCM

URI: nescm/test/uri

Password expiration servlet:

Authentication Level: 0

☐ Satisfiable by a contract of equal or higher level

☐ Satisfiable by External Provider

If you add more than one X509 method, only the first one will be used and it will automatically be moved to the top of the list.

Methods: Method-NMAS-NESCM

Available methods: Name/Password - Basic, Name/Password - Form, Secure Name/Password - Basic, Secure Name/Password - Form

2 Specify a *Display name* (for example, `Contract-NMAS-NESCM-UserStore1`).

3 Enter a *URI* (for example, `nescm/test/uri`).

The URI is used to identify this contract for external providers and is a unique path value that you create.

4 In the *Available methods* list, select the method created in "Creating a Method to Use the NMAS Class" on page 133, then click the left-arrow to move this method into the *Methods* list.

All other fields can remain in the default state.

5 Click *Finish*, then click *OK*.

6 Update the Identity Server by clicking *Update*.

7 Update the Access Gateway.

8 Continue with Section 8.10.4, "Assigning the NESCM Contract to a Protected Resource," on page 135

8.10.4 Assigning the NESCM Contract to a Protected Resource

Contracts must be created before they can be assigned to protected resources. The following steps explain how to assign the NESCM contract to an existing protected resource. If you have not created a protected resource, see the *Novell Access Manager Setup Guide* (http://www.novell.com/documentation/novellaccessmanager/basicconfig/data/bookinfo.html).

1 In the Administration Console, click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.

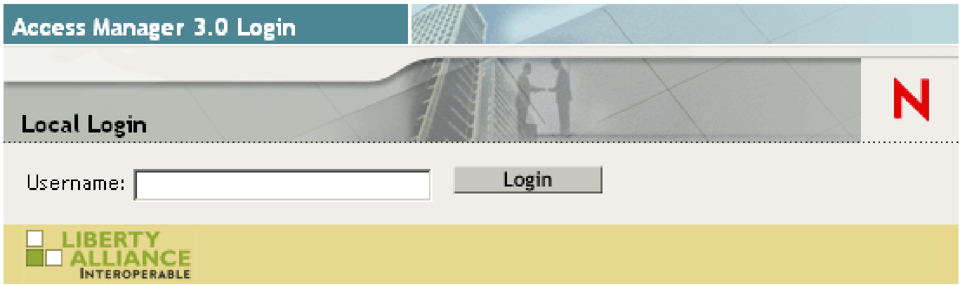
The reverse proxy should be configured with a resource that you want to protect with the smart card.

2. Click the *Protected Resource* link for the accelerator where you want to assign the NESCM contract.
3. To enable the NESCM contract on an existing protected resource, click the *Contract* link for that resource, then in the *Contract* selection list, select the NESCM contract created in "Creating an Authentication Contract to Use the Method" on page 134.

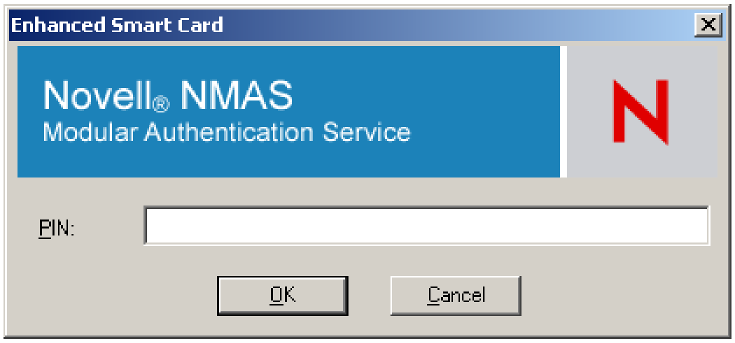
   If the contract is not listed, make sure you have updated both the Identity Server and the Access Gateway, in this specified order. If you have multiple Identity Server configurations, make sure that the Access Gateway is assigned to the Identity Server configuration that contains the NESCM contract (click *Access Gateways* > *Edit* > *Reverse Proxy / Authentication*).
4. Click *OK*.
5. Click the *Access Gateways* task, then update the Access Gateway.
6. Continue with Section 8.10.5, "Verifying the User's Experience," on page 136.

8.10.5 Verifying the User's Experience

1. From the smart-card-equipped workstation, browse to and select the URL of the accelerator where the protected resource requiring NESCM type authentication is enabled.
2. When prompted by Access Manager, enter a *username*.



3. When prompted for the smart card password, enter a password (the smart card PIN).



If the Smart Card contains a certificate that meets the defined criteria (in this example, a matching Subject name and trusted signing CA), the user is now successfully authenticated to the IDP and is connected through the Access Gateway to the protected resource.

8.10.6 Troubleshooting

Error	Resolution
Authentication fails without prompting the user for the token	Verify that you have configured the class and method correctly. See "Creating an NMAS Class for NESCM" on page 132 and "Creating a Method to Use the NMAS Class" on page 133
Certificate validation fails	Verify that a trusted root object created for the signing CA of the certificate on the Smart Card exists in the eDirectory trusted root container

Configuring Trusted Providers

9

This section discusses configuring trust so that two user accounts can be associated with each other without the sites exchanging data. It explains how to set up the trust with internal and external trusted identity providers, service providers, and embedded service providers (ESPs). Steps for configuring trusted provider types are similar, and are also similar between the Liberty and SAML protocols. The interface pages in this section show the configuration of a Liberty trusted service provider.

- Section 9.1, "Understanding the Trust Model," on page 139
- Section 9.2, "Creating a Trusted Provider Reference," on page 142
- Section 9.3, "Reimporting a Trusted Provider's Metadata," on page 144
- Section 9.4, "Configuring General Provider Options," on page 144
- Section 9.5, "Editing a SAML 1.1 Trusted Identity Provider's Metadata," on page 146
- Section 9.6, "Editing a SAML 1.1 Trusted Service Provider's Metadata," on page 148
- Section 9.7, "Configuring Common Access Settings for a Trusted Provider," on page 149
- Section 9.8, "Selecting Attributes for a Trusted Provider," on page 155

About SAML and Liberty

For information about how Access Manager uses SAML, see Appendix B, "Understanding How Access Manager Uses SAML," on page 699.

For conceptual information about Liberty, see Appendix A, "About Liberty," on page 697.

For troubleshooting information, see Chapter 38, "Troubleshooting for the Identity Server and Authentication," on page 587.

9.1 Understanding the Trust Model

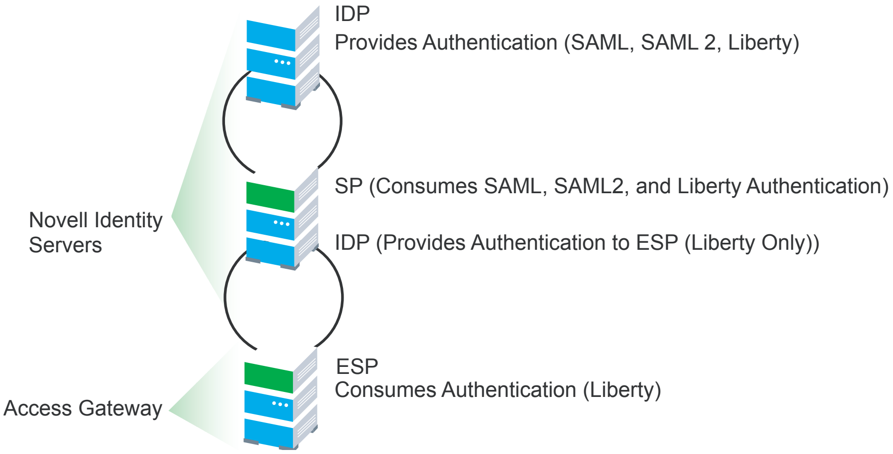
Setting up trust involves system administrators agreeing on how to establish a secure method for providing and consuming authentication assertions between their Identity Servers. An Identity Server is always installed as an identity provider, which is used to provide authentication to trusted service providers and embedded service providers (ESPs).

- Section 9.1.1, "Identity Consumer," on page 139
- Section 9.1.2, "Embedded Service Providers," on page 140
- Section 9.1.3, "High-Level Steps," on page 141

9.1.1 Identity Consumer

An Identity Server also can be configured as an identity consumer (service provider), which enables the Identity Server to consume authentication assertions from trusted identity providers. Figure 9-1 depicts how two Identity Servers can be configured in a trust model using the SAML and Liberty protocols to provide authentication for an Access Gateway ESP.

Figure 9-1 *Identity Server Trust*

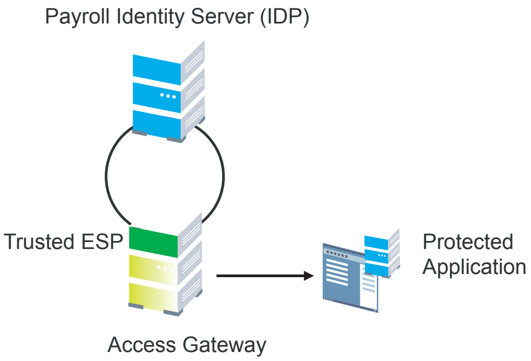

IDP
Provides Authentication (SAML, SAML 2, Liberty)
SP (Consumes SAML, SAML2, and Liberty Authentication)
Novell Identity Servers
IDP (Provides Authentication to ESP (Liberty Only))
ESP
Consumes Authentication (Liberty)
Access Gateway

As an administrator, you determine whether your server is to be used as the identity provider or service provider in the trust relationship. You and the trusted partner agree to exchange Identity Server metadata, and then you create references to the trusted partner's Identity Server in your Identity Server configuration. You can obtain metadata via a URL or an XML document, then enter it in the system when you create the reference.

9.1.2 Embedded Service Providers

In addition to setting up trust with internal or external service providers, you can reference embedded service providers (ESPs) in your enterprise. An ESP uses the Liberty protocol and does not require metadata entry, because this exchange happens automatically. The ESP comes with Access Manager and is embedded in the Access Gateway and the J2EE agent. The ESP facilitates authentication between the Identity Server and the resource protected by the Access Gateway or agent, as shown in as shown in Figure 9-2.

Figure 9-2 *Embedded Service Provider*



The components in this example reside in the same trust store and represent a typical Access Manager configuration used within an enterprise.

9.1.3 High-Level Steps

The following high-level steps describe setting up the trust model between an identity provider and a service provider. These steps assume that both providers are using the Novell® Identity Server provided with Access Manager.

1. Administrators at each company install and configure the Identity Server.

   The Identity Server that consumes authentications must be enabled to run as a service provider. See Section 6.1.1, “Creating a Cluster Configuration,” on page 56. (It is recommended that you are already familiar with the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.)

2. Administrators at each company must import the trusted root certificate of the other Identity Server into the NIDP trust store.

   Click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Security* > *NIDP Trust Store*, then auto import the certificate. Use the SSL port (8443) even if you haven’t set up the base URL of the Identity Server to use HTTPS.

3. Administrators must exchange Identity Server metadata with the trusted partner.

   Metadata is generated by the Identity Server and can be obtained via a URL or an XML document, then entered in the system when you create the reference. This step is not applicable if you are referencing an ESP. When you reference an ESP, the system lists the installed ESPs for you to choose, and no metadata entry is required.

4. Create the reference to the trusted identity provider and the service provider.

   This procedure associates the metadata with the new provider. See Section 9.2, “Creating a Trusted Provider Reference,” on page 142.

5. Configure user authentication.

   This procedure defines how your Identity Server interacts with the trusted provider during user authentication. Access Manager comes with default basic authentication settings already enabled. See Chapter 10, “Configuring User Authentication and Federation,” on page 157.

   Additional important steps for enabling authentication between trusted providers include:

   - Setting up the necessary authentication contracts. See Section 8.4, “Configuring Authentication Contracts,” on page 111.
   - Enabling the profiles that you are using. See Section 12.2, “Enabling Web Services and Profiles,” on page 174.
   - Enabling the *Always Allow Interaction* option on the Web Service Consumer page. See Section 12.8, “Configuring the Web Service Consumer,” on page 184.

6. (Conditional) If you are setting up SAML 1.1 federation, the protocol does not allow the federation link to be automatically added to the login page. You must manually configure this setting.

   See Section 9.7.1, “Configuring Display and Access Settings,” on page 149. Specify a value for the *Login URL* and *Destination URL*, then select *Advertise (Display) on Login Dialog*.

   For more information, see TID 3247813 (https://secure-support.novell.com/KanisaPlatform/Publishing/992/3247813_f.SAL_Public.html).

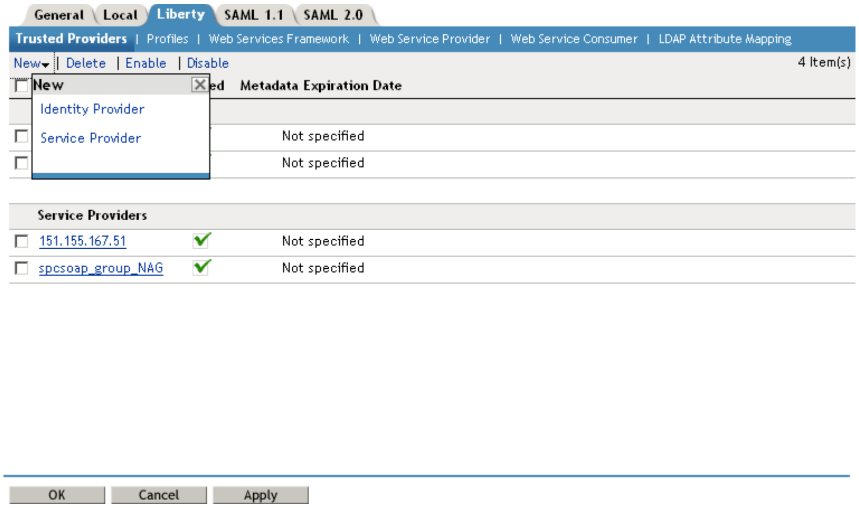
9.2 Creating a Trusted Provider Reference

The procedure for establishing trust between providers begins with obtaining metadata for the trusted provider. If you are using the Novell Identity Server, protocol-specific metadata is available via a URL. Examples of metadata URLs for server 10.1.1.1 would be:

- **Liberty:** http://10.1.1.1:8080/nidp/idff/metadata
- **SAML 1.1:** http://10.1.1.1:8080/nidp/saml/metadata
- **SAML 2.0:** http://10.1.1.1:8080/nidp/saml2/metadata

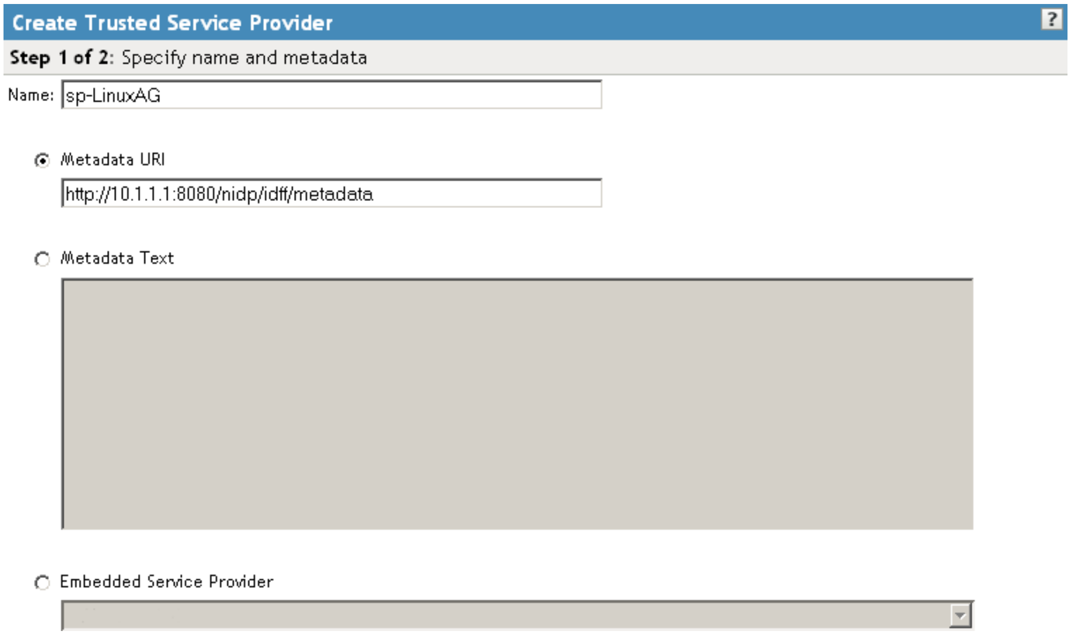
The default values nidp and 8080 are established during product installation; nidp is the Tomcat application name.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > [Protocol]*.


General
Local
Liberty
SAML 1.1
SAML 2.0
Trusted Providers | Profiles | Web Services Framework | Web Service Provider | Web Service Consumer | LDAP Attribute Mapping
New | Delete | Enable | Disable
4 Item(s)
New
Identity Provider
Service Provider
OK
Cancel
Apply

☐		ed	Metadata Expiration Date
☐			Not specified
☐			Not specified
	Service Providers		
☐	151.155.167.51	✓	Not specified
☐	spcsoap_group_NAG	✓	Not specified

2 Click *New*, then click *Identity Provider* or *Service Provider*.



Create Trusted Service Provider

Step 1 of 2: Specify name and metadata

Name: sp-LinuxAG

● Metadata URI

http://10.1.1.1:8080/nidp/idff/metadata

○ Metadata Text

○ Embedded Service Provider

3 In the Name option, specify a name by which you want to refer to the provider.

4 Select one of the following methods to obtain the metadata:

Metadata URL: Specify the metadata URL for a trusted provider. The system retrieves protocol metadata using the specified URL.

If your Identity Server and Administration Console are on different machines, use HTTP to import the metadata. If you are required to use HTTPS with this configuration, you must import the trusted root certificate of the provider into the trust store of the Administration Console. You need to use the Java `keytool` to import the certificate into the `cacerts` file in the `/opt/novell/java/jre/lib/security` directory of the Administration Console.

If you do not want to use HTTP and you do not want to import a certificate into the Administration Console, you can use the *Metadata Text* option. In a browser, enter the HTTP URL of the metadata. View the text from the source page, save the source metadata, then paste it into the *Metadata Text* option.

Metadata Text: An editable field in which you can paste copied metadata text from an XML document, assuming you obtained the metadata via e-mail or disk and are not using a URL. If you copy metadata text from a Web browser, you must copy the text from the page source.

Manual Entry: (SAML 1.1 only) Allows you to enter metadata values manually. When you select this option, the system displays the Enter Metadata Values page. See Section 9.5, "Editing a SAML 1.1 Trusted Identity Provider's Metadata," on page 146.

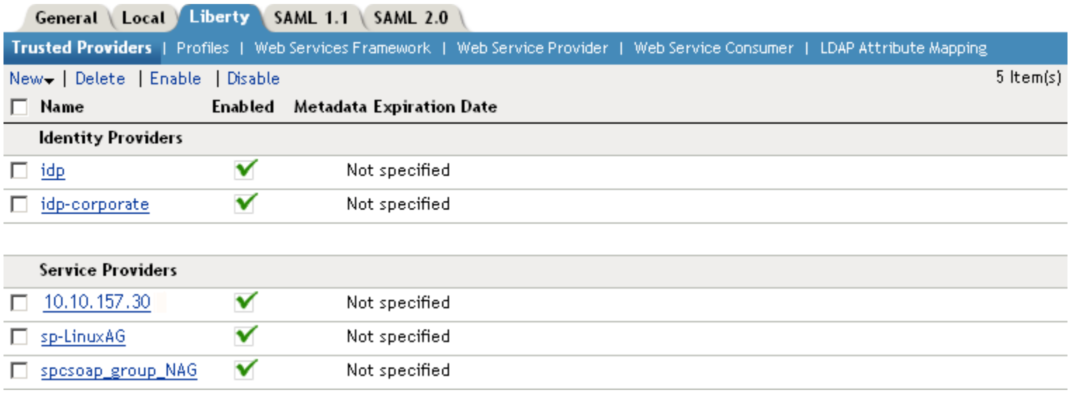
5 If you are creating a service provider for an Access Gateway or agent, click the following option:

Embedded Service Provider: Access Gateway and application server agents (J2EE or Windows) include an embedded service provider (ESP) that can be trusted by identity providers. ESPs run in the same enterprise as the identity provider, and are therefore created and configured in the same directory. The ESP enables all of the single-sign on functionality for Access Gateway or agent. Installed ESPs are displayed in a drop-down list for you to select as a trusted entity. You do not need to enter metadata for an ESP; it is automatically generated.

6 Click *Next*.

7 Review the metadata certificates, then click *Finish*.

8 The system displays the trusted provider on the Liberty page.


General
Local
Liberty
SAML 1.1
SAML 2.0
Trusted Providers | Profiles | Web Services Framework | Web Service Provider | Web Service Consumer | LDAP Attribute Mapping
New | Delete | Enable | Disable
5 Item(s)

☐ Name	Enabled	Metadata Expiration Date
Identity Providers		
☐ idp	✓	Not specified
☐ idp-corporate	✓	Not specified
Service Providers		
☐ 10.10.157.30	✓	Not specified
☐ sp-LinuxAG	✓	Not specified
☐ spcsoap_group_NAG	✓	Not specified

9.3 Reimporting a Trusted Provider's Metadata

You might need to reimport a trusted provider's metadata if you learn that it has changed, or if you have changed the base URL of the Identity Server configuration. The steps to do this are similar for Liberty and SAML protocols.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *[Liberty or SAML]* > *[Trusted Provider]* > *Metadata*.

2 Click the trusted provider, then click the *Metadata* tab.

3 Click *Reimport*.

Follow the prompts to import the metadata.

4 Specify the new metadata information as described in Section 9.2, "Creating a Trusted Provider Reference," on page 142.

5 Confirm metadata certificates, then click *Finish*.

9.4 Configuring General Provider Options

The following options are global in that they affect any identity provider or identity consumer that the Identity Server has been configured to trust:

- Section 9.4.1, "Configuring the General Identity Provider Options," on page 144
- Section 9.4.2, "Configuring the General Identity Consumer Options," on page 145

9.4.1 Configuring the General Identity Provider Options

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.

2 To specify identity provider settings, fill in the following fields:

Show logged out providers: Displays logged-out providers on the identity provider's log-out confirmation page.

Require Signed Authentication Requests: Specifies that for the Liberty 1.2 and SAML 2.0 protocols, authentication requests from service providers must be signed. When you enable this option for the identity provider, you must also enable the *Sign Authentication Requests* option under the *Identity Consumer* heading on this page for the external trusted service provider. (It is possible, however, to configure an identity provider that requires signed requests to function as an identity consumer that does not sign requests.)

Use Introductions (Publish Authentications): Enables single sign-on from the service provider to the identity provider. The service provider determines the identity providers that users are already logged into, and then selectively and automatically asks for authentication from one of the identity providers. Introductions are enabled only between service and identity providers that have agreed to a circle of trust, which means that they have agreed upon a common domain name for this purpose.

After authenticating a user, the identity provider accesses a service at the service domain and writes a cookie to the common part of the service domain, publishing that the authentication has occurred.

- **Service Domain (Local and Common):** Enables a service provider to access a service at the service domain prior to authenticating a user. This service reads cookies obtained at this domain and discovers if any identity providers have provided authentication to the user. The service provider determines whether any of these identity providers can authenticate a user without credentials. The service domain must resolve to the same IP address as the base URL domain.

  For example, if an agreed-upon common domain is *xyz.com*, the service provider can specify a service domain of *sp.xyz.com*, and the identity provider can specify a service domain of *idp.xyz.com*. For the identity provider, *xyz.com* is the common value entered, and *idp* is the local value.
- **Port:** The port to use for identity provider introductions. Port 8445 for HTTPS is the default and must be opened on your firewall. If you specify a different port, you must edit the Tomcat server XML.

SSL Certificate: Displays the Keystore page that you use to locate and replace the test-provider SSL certificate for this configuration.

The Identity Server comes with a test-provider certificate that you must replace for your production environment. This certificate is used for identity provider introductions. You can replace the test certificate now or after you have configured the Identity Server. If you create the certificate and replace the test-connector now, you can save some time by restarting Tomcat only once. Tomcat must be restarted whenever you assign an Identity Server to a configuration and whenever you update a certificate key store. See Section 6.5.3, “Managing the Keys, Certificates, and Trust Stores,” on page 80.

3 Click *OK*, then update the Identity Server.

9.4.2 Configuring the General Identity Consumer Options

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.

2 To specify whether the Identity Server also runs as an identity consumer.

If configured to run as an identity consumer, the Identity Server can receive (consume) authentication assertions from other identity providers.

Enable: Enables this site to function as service provider. This setting is enabled by default.

Require Signed Assertions: Specifies that all SAML assertions received by the service provider must be signed by the issuing SAML authority. The signing authority uses a key pair to sign SAML data sent to this trusted provider.

Sign Authentication Requests: Specifies that the service provider signs authentication requests to an identity provider for the Liberty 1.2 and SAML 2.0 protocols.

Use Introductions (Discover IDP Authentications): Enables a service provider to discover whether a user has authenticated to a trusted identity provider, so the user can use single sign-on without requiring authentication credentials.

- **Service domain:** The shared, common domain for all providers in the circle of trust. This domain must resolve to the same IP address as the base URL domain. You must enable the *Identity Consumer* option to enable this field.
- **Port:** The port to use for identity consumer introductions. Port 8446 for HTTPS is the default and must be opened on your firewall. If you specify a different port, you must edit the Tomcat server XML.

SSL Certificate: Displays the Keystore page that you use to locate and replace the test-consumer SSL certificate for this configuration.

The Identity Server comes with a test-consumer certificate that you must replace for your production environment. This certificate is used for identity consumer introductions. You can replace the test certificate now or after you have configured the Identity Server. If you create the certificate and replace the test-connector now, you can save some time by restarting Tomcat only once. Tomcat must be restarted whenever you assign an Identity Server to a configuration and whenever you update a certificate key store. See Section 6.5.3, "Managing the Keys, Certificates, and Trust Stores," on page 80.

3 Click *OK*, then update the Identity Server.

9.5 Editing a SAML 1.1 Trusted Identity Provider's Metadata

Access Manager allows you to obtain metadata for SAML 1.1 providers. However, metadata for SAML 1.1 might not be available for some trusted providers. Therefore, you can enter metadata manually. The page for this is available if you clicked the *Manual Entry* option when you created the trusted provider.

IMPORTANT: The SAML 2.0 and Liberty 1.2 protocols define a logout mechanism whereby the service provider sends a logout command to the trusted identity provider when a user logs out at a service provider. SAML 1.1 does not provide such a mechanism. For this reason, when a log-out occurs at the SAML 1.1 service provider, no log-out occurs at the trusted identity provider. A valid session is still running at the identity provider, and no credentials need to be entered. In order to log out at both providers, the user must navigate to the identity provider that authenticated him to the SAML 1.1 service provider and log out manually.

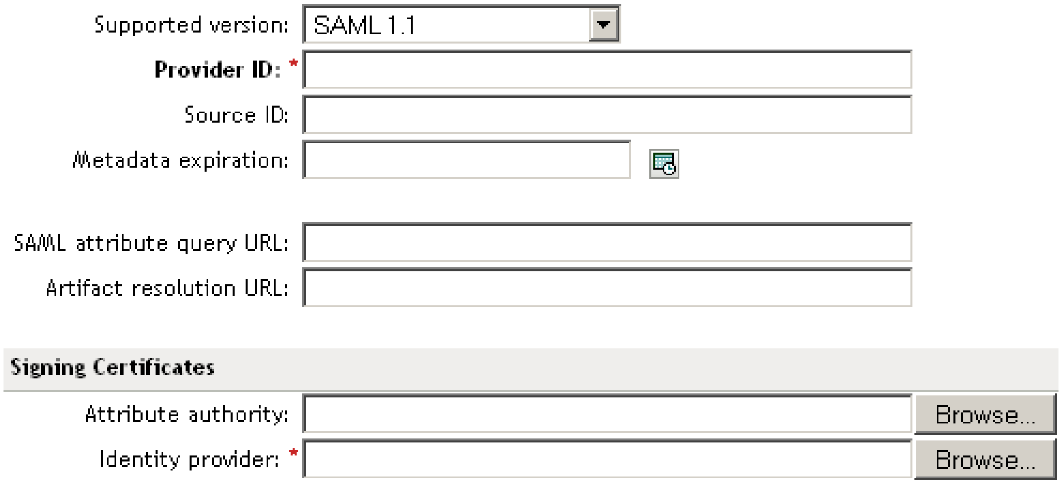
For conceptual information about how Access Manager uses SAML, see Appendix B, "Understanding How Access Manager Uses SAML," on page 699.

1 In the Administration Console, click *Access Manager > Identity Servers > Edit > SAML 1.1 > [Identity Provider] > Metadata*.

2 To reimport the metadata from a URL or text, click *View*, then click *Reimport*.

The system displays the Create Trusted Identity Provider Wizard that lets you obtain the metadata. Follow the on-screen instructions to complete the steps in the wizard.

3 To edit the metadata manually, click *Edit*.


Supported version: SAML 1.1

4 Fill in the following fields as necessary:

Supported Version: Specifies the version of SAML that you want to use.

Provider ID: (Required) The SAML 1.1 metadata unique identifier for the provider. For example, https://dns.name:port/nidp/saml/metadata.

Source ID: The SAML Source ID for the trusted provider. The Source ID is a 20-byte value that is used as part of the Browser/Artifact profile. It allows the receiving site to determine the source of received SAML artifacts. If none is specified, the Source ID is auto-generated using a SHA-1 hash of the site provider ID.

Metadata expiration: The date upon which the metadata is no longer valid.

SAML attribute query URL: The URL location where an attribute query is to be sent to the partner. The attribute query requests a set of attributes associated with a specific object. A successful response contains assertions that contain attribute statements about the subject. A SAML 1.1 provider might use the base URL, followed by /saml/soap. For example, https://[dns:port]/nidp/saml/soap.

Artifact resolution URL: The URL location where artifact resolution queries are sent. A SAML artifact is included in the URL query string. The target URL on the destination site the user wants to access is also included on the query string. A SAML 1.1 provider might use the base URL, followed by /saml/soap. For example, https://[dns:port]/nidp/saml/soap.

5 To specify signing certificate settings, fill in the following fields:

Attribute authority: Specifies the signing certificate of the partner SAML 1.1 attribute authority. The attribute authority relies on the identity provider to provide it with authentication information so that it can retrieve attributes for the appropriate entity or user. The attribute authority must know that the entity requesting the attribute has been authenticated to the system.

Identity provider: (Required) Appears if you are editing identity provider metadata. This field specifies the signing certificate of the partner SAML 1.1 identity provider. It is the certificate the partner uses to sign authentication assertions.

6 Click *OK*.

7 On the Identity Servers page, click *Update All* to update the configuration.

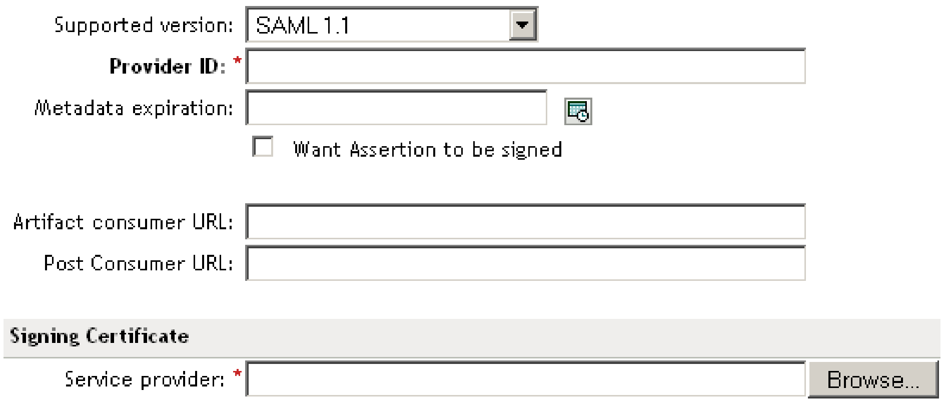
9.6 Editing a SAML 1.1 Trusted Service Provider's Metadata

Access Manager allows you to obtain metadata for SAML 1.1 providers. However, metadata for SAML 1.1 might not be available for some trusted providers. Therefore, Access Manager allows you to enter metadata manually. The page for this is available if you clicked the *Manual Entry* option when you created the trusted provider.

For conceptual information about how Access Manager uses SAML, see Appendix B, "Understanding How Access Manager Uses SAML," on page 699.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *SAML 1.1* > *[Service Provider]* > *Metadata*.
2. If you want to reimport the metadata, click *View*, then click *Reimport*.

   Follow the on-screen instructions to complete the steps in the wizard.
3. Click *Edit*.



Supported version: SAML 1.1

Provider ID: *

Metadata expiration:

☐ Want Assertion to be signed

Artifact consumer URL:

Post Consumer URL:

Signing Certificate

Service provider: *

4. Fill in the following fields:

   Supported Version: Specifies which version of SAML that you want to use.

   Provider ID: (Required) Specifies the SAML 1.1 metadata unique identifier for the provider. For example, https://dns.name:port/nidp/saml/metadata.

   Metadata expiration: Specifies the date upon which the metadata is no longer valid.

   Want assertion to be signed: Specifies that authentication assertions from the trusted provider must be signed.

   Artifact consumer URL: Specifies where the partner receives incoming SAML artifacts. For example, https://[dns:port]/nidp/saml/spassertion_consumer.

   Post consumer URL: Specifies where the partner receives incoming SAML POST data. For example, https://[dns:port]/nidp/saml/spassertion_consumer.

   Service Provider: Specifies the public key certificate used to sign SAML data. You can browse to locate the service provider certificate.
5. Click *Finish*.

9.7 Configuring Common Access Settings for a Trusted Provider

Common access settings for a trusted provider include specifying how to display the provider on the Identity Server's login page, and specifying security settings for communication.

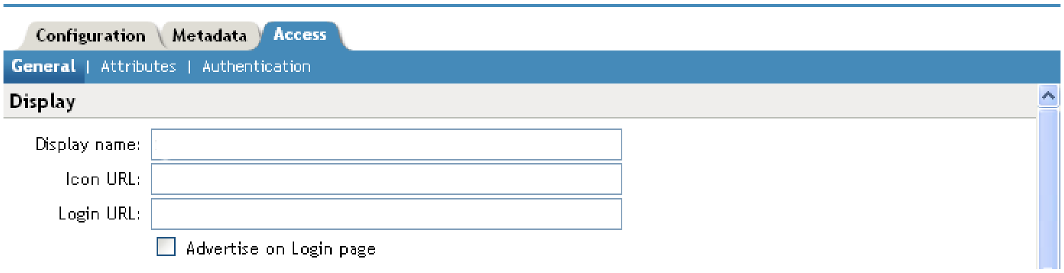
- Section 9.7.1, "Configuring Display and Access Settings," on page 149
- Section 9.7.2, "Configuring Communication Security Settings," on page 150
- Section 9.7.3, "The Intersite Transfer Service," on page 151

9.7.1 Configuring Display and Access Settings

You can configure how you want to display the provider on the Identity Server's login page. The fields that are displayed on this page vary depending on the protocol and provider type you selected for configuration.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > [Protocol] > [Provider Name]*.

2 Click *Access > General*.



Display

Display name:

Icon URL:

Login URL:

☐ Advertise on Login page

3 Fill in the following fields:

Display name: The display name seen by the end user for this trusted provider. The default name is the name you entered when creating the trusted provider.

Icon URL: The URL of the icon to display for this trusted provider. If you add an icon, the system displays the icon as the link, rather than the text in the *Display name* field.

Login URL: (Displayed for a SAML 1.1 trusted identity provider) The URL required by the identity provider to authenticate the user from the service provider.

For Liberty and SAML 2.0, a single-sign on URL is automatically generated and can be displayed on the service provider's login page as a link to the identity provider. For SAML 1.1, you must specify an Intersite Transfer Service URL in the *Login URL* field. For more information, see Section , "Specifying the Intersite Transfer Service URL for the Login URL Option," on page 153.

Destination URL: (Displayed for a SAML 1.1 trusted service provider) This setting is not currently used and can be left blank.

Advertise on Login page: Displays the identity provider's link on the Login page when the user logs directly into the User Portal (...nidp/app).

4 Click *OK*.

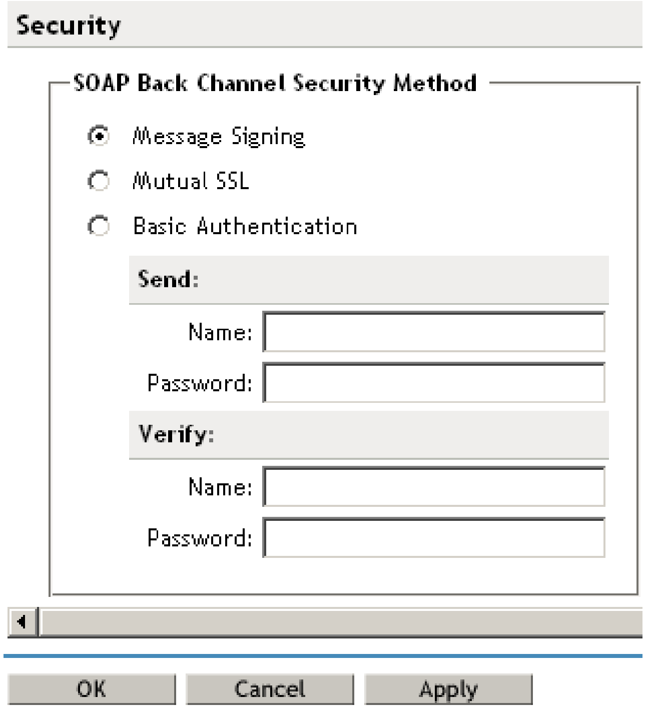
5 Click *OK* on the Trusted Providers page.

6 Click *Update Servers* on the Servers page.

9.7.2 Configuring Communication Security Settings

You can configure the security settings to control direct communication between the Identity Server and a trusted provider across the SOAP back channel. These methods apply to the trusted identity provider and are similar between Liberty and SAML.

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > [Protocol] > [Provider Name]*.
2. Click *Access > General*.



● Message Signing

○ Mutual SSL

○ Basic Authentication

3. Fill in the following fields:

   Encrypt assertions: (SAML 2.0 security only). Adds a level of security by encrypting assertions so that they cannot be meaningfully interpreted at an intermediate entity.

   Encrypt name identifiers: (SAML 2.0 identity provider security). Adds a level of security by encrypting name identifiers so that they cannot be meaningfully interpreted at an intermediate entity.

   The following settings specify how to validate messages received from trusted providers over the SOAP back channel.

   Message Signing: Specifies no security but rather relies upon message signing using a digital signature.

   Mutual SSL: Specifies that this trusted provider provides a digital certificate (mutual SSL) when it sends a SOAP message.

   SSL communication requires only the client to trust the server. For mutual SSL, the server must also trust the client. For the client to trust the server, the server's certificate authority (CA) certificate must be imported into the client trust store. For the server to trust the client, the client's certificate authority (CA) certificate must be imported into the server trust store.

Basic Authentication: Standard header-based authentication. This method assumes that a name and password for authentication are sent and received over the SOAP back channel.

Send: The name and password to be sent for authentication to the trusted partner. The partner expects this password for all SOAP back-channel requests, which means that the name and password must be agreed upon.

Verify: The name and password used to verify data that the trusted provider sends.

4. Click *OK*.
5. Click *OK* on the Trusted Providers page.
6. Click *Update Servers* on the Servers page.

9.7.3 The Intersite Transfer Service

- "Understanding the Intersite Transfer Service URL" on page 151
- "Using Intersite Transfer Service Links on Web Pages" on page 152
- "Specifying the Intersite Transfer Service URL for the Login URL Option" on page 153

Understanding the Intersite Transfer Service URL

The Intersite Transfer Service is used by an identity provider to cause authentication to occur at a service provider that it trusts. The URLs for accessing the Intersite Transfer Service are different for each supported protocol (Liberty, SAML 1.1, and SAML 2.0). The Novell Access Manager identity and service provider components use the following format of the Intersite Transfer Service URL:

- **SAML 1.1:** <*identity_provider_base_URL*>/saml/idpsend?
PID=<*service_provider_base_URL*>/nidp/saml/metadata&
TARGET=<*final_destination_URL*>
- **SAML 2.0:** <*identity_provider_base_URL*>/saml2/idpsend?
PID=<*service_provider_base_URL*>/nidp/saml2/metadata&
TARGET=<*final_destination_URL*>
- **Liberty:** <*identity_provider_base_URL*>/idff/idpsend?
PID=<*service_provider_base_URL*>/nidp/idff/metadata&
TARGET=<*final_destination_URL*>

The <*identity_provider_base_URL*> is the Base URL of the identity provider that is providing authentication, followed by the path to the protocol application being used for federation. Notice that the path is different for each protocol.

The <*service_provider_base_URL*> is the Base URL of the service provider, followed by the path to the protocol metadata. Notice that the path is different for each protocol. The scheme (http or https) in the PID must match what is configured for the Base URL for the service provider.

The <*final_destination_URL*> is the URL to which the browser is redirected following a successful authentication at the identity provider. If this target URL contains parameters (for example, TARGET=https://login.provo.novell.com:8443/nidp/app?function_id=22166&amp;Resp_Id=55321&amp;Resp_App_Id=810&amp;security_id=0), it must be URL encoded to prevent the URL from being truncated.

Examples:

- **SAML 1.1:** `https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://eng.provo.novell.com/saml1/myapp`
- **SAML 2.0:** `https://idp.sitea.novell.com:8443/nidp/saml2/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml2/metadata&TARGET=https://eng.provo.novell.com/saml2/myapp`
- **Liberty:** `https://idp.sitea.novell.com:8443/nidp/idff/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/idff/metadata&TARGET=https://eng.provo.novell.com/liberty/myapp`

The Intersite Transfer Service URLs of third-party identity and service provider implementations are different than those shown above for the Novell providers. Check the third party documentation for the URL information.

Using Intersite Transfer Service Links on Web Pages

The Intersite Transfer Service URL can be used on a Web page that provides links to various protected resources requiring authentication with a specific identity provider and a specific protocol. Links on this Web page are configured with the URL of the Intersite Transfer Service of the identity provider to be used for authentication. Clicking these links directs the user to the appropriate identity provider for authentication. Following successful authentication, the identity provider sends a SAML assertion to the service provider. The service provider uses the SAML assertion to verify authentication, and then redirects the user to the destination URL as specified in the TARGET portion of the Intersite Transfer Service URL.

Below are sample links that might be included on a Web page. These links demonstrate the use of SAML 1.1, SAML 2.0, and Liberty formats for the Intersite Transfer Service URL:

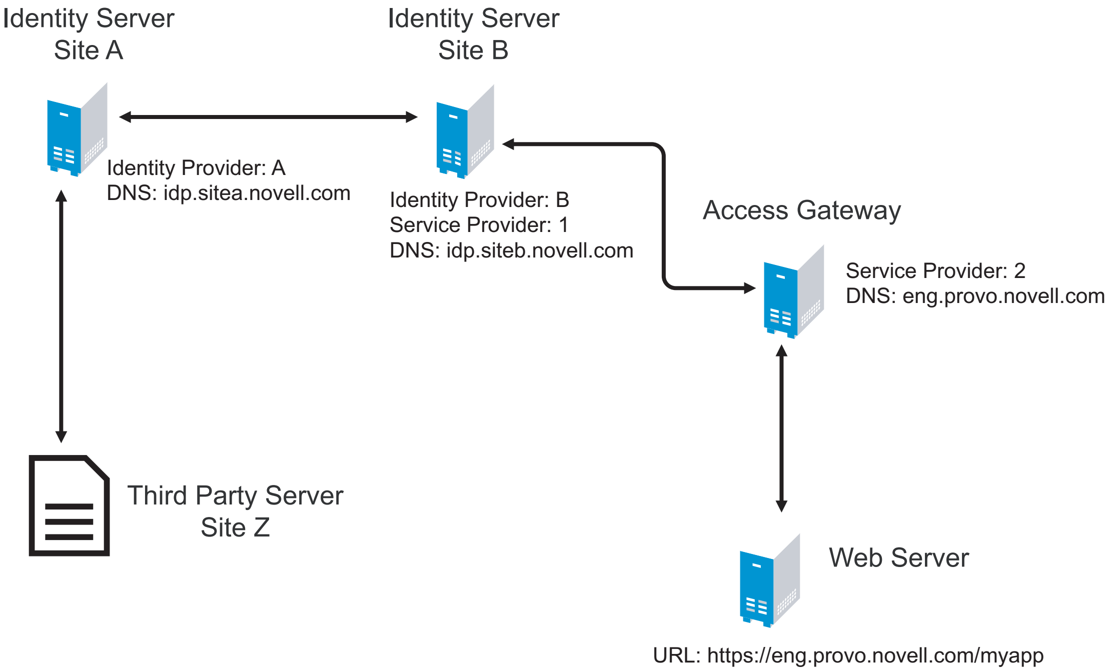
SAML 1.1: `<a href="https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://eng.provo.novell.com/saml1/myapp">SAML1 example</a> <br> <br>`

SAML 2.0: `<a href="https://idp.sitea.novell.com:8443/nidp/saml2/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/saml2/metadata&TARGET=https://eng.provo.novell.com/saml2/myapp">SAML2 example</a> <br> <br>`

Liberty: `<a href="https://idp.sitea.cit.novell.com:8443/nidp/idff/idpsend?PID=https://idp.siteb.novell.com:8443/nidp/idff/metadata&TARGET=https://eng.provo.novell.com/liberty/myapp">Liberty example</a> <br> <br>`

Figure 9-3 illustrates a network configuration that could use these sample links.

Figure 9-3 *Using the Intersite Transfer Service URL*

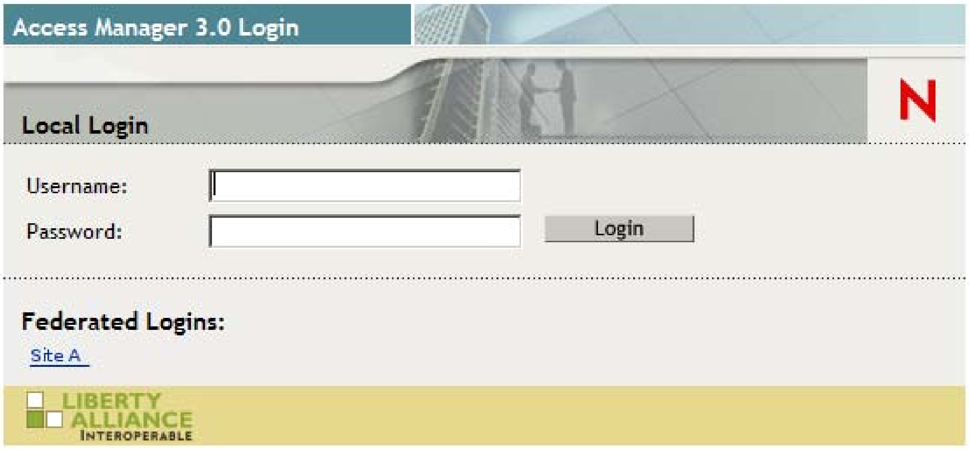


In this example, Site Z places links on its Web page using the Intersite Transfer Service URL of Site A. These links trigger authentication at Site A. If successful, Site A sends an assertion to Site B. Site B verifies the authentication and redirects the user the myapp application that it is protecting.

Specifying the Intersite Transfer Service URL for the Login URL Option

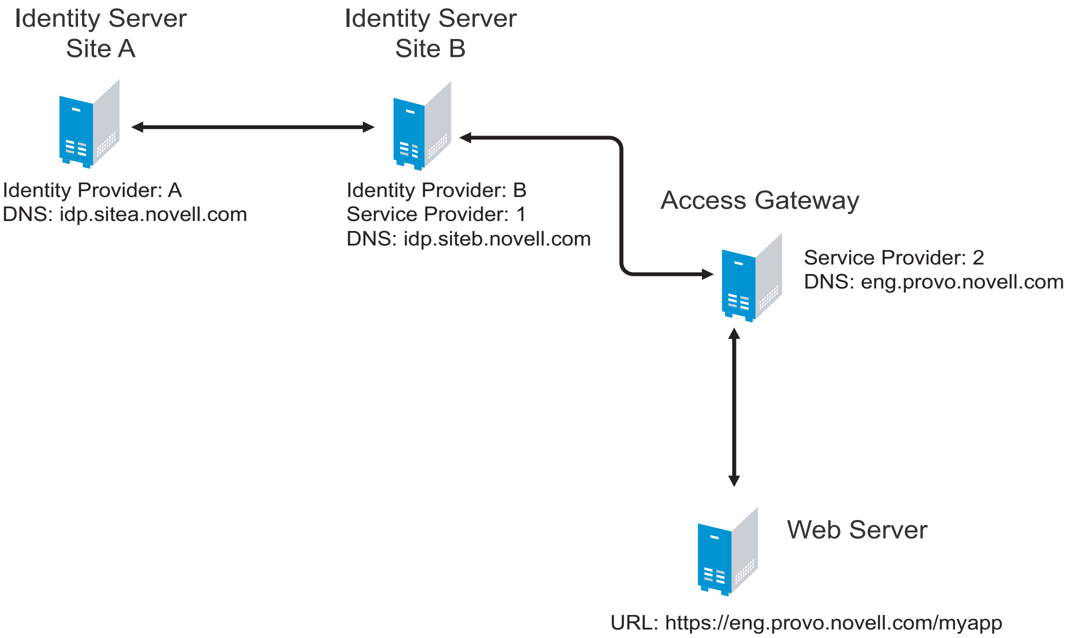
The Liberty and SAML 2.0 protocols allow the Identity Server to create single sign-on links on the Login page for the User Portal (... /nidp/app) when you select the *Advertise on Login page* option (see Section 9.7.1, "Configuring Display and Access Settings," on page 149). The link on the Login page looks similar to the following:

Figure 9-4 *Federated Links on the Login Page*



The SAML 1.1 protocol does not support a single sign-on URL. In order for a link to appear on the Login page, you must specify a Intersite Transfer Service URL in the *Login URL* option. Figure 9-5 illustrates a possible configuration that requires the Intersite Transfer Service for the SAML 1.1 protocol.

Figure 9-5 *Federated Identity Configuration*



If you want a link to appear on the login page that allows the user to log in to Site A (as shown in Figure 9-4), you need to specify a value for the *Login URL* option.

Using the DNS names from Figure 9-5, the complete value for the *Login URL* option is as follows:

```
https://idp.sitea.novell.com:8443/nidp/saml/idpsend?PID=https://
idp.siteb.novell.com:8443/nidp/saml/metadata&TARGET=https://
idp.siteb.novell.com:8443/nidp/app
```

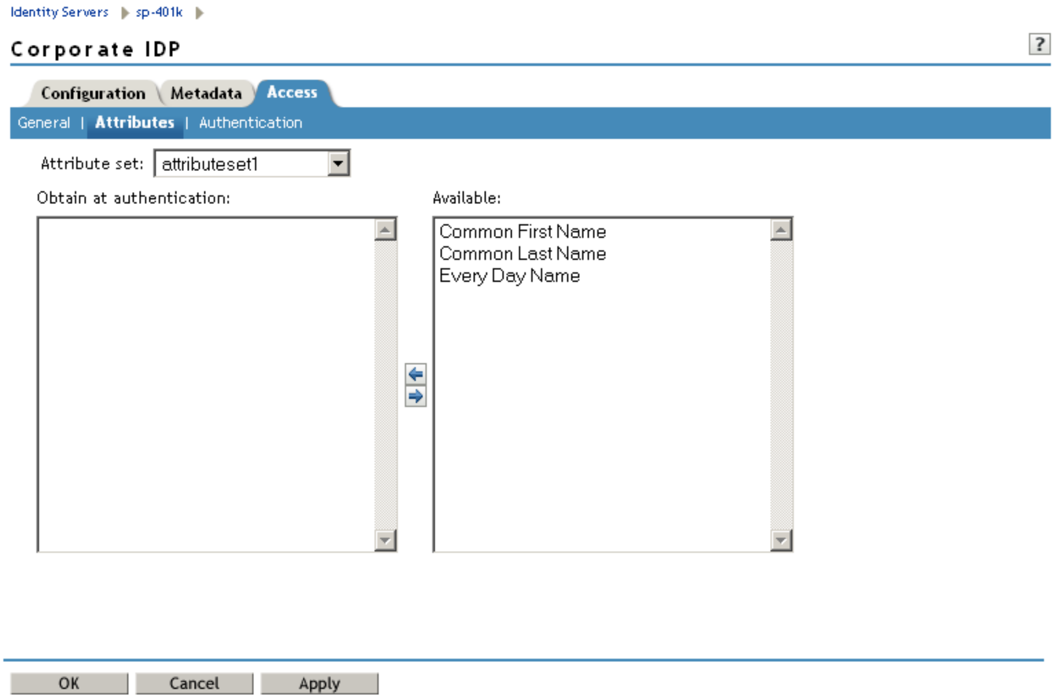
The following happens when this link is invoked:

1. The browser performs a Get to the identity provider (Site A).
2. If the identity provider (Site A) trusts the service provider (Site B), the identity provider prompts the user for authentication information and builds an assertion.
3. The identity provider (Site A) sends the user to the service provider (Site B) using the POST or Artifact method.
4. The service provider (Site B) consumes the assertion and sends the user to the TARGET URL (the user portal on Site B).

9.8 Selecting Attributes for a Trusted Provider

You can select attributes that an identity provider sends and a service provider receives in an authentication. You can also create attribute sets or select attribute sets that you created globally in Section 7.1, “Configuring Attribute Sets,” on page 83.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty [or SAML]* > *[Provider]* > *Access* > *Attributes*.



Attribute set: attributeset1

2 To create an attribute set, select *New Attribute Set* from the *Attribute Set* drop-down menu.

An attribute set is a group of attributes that can be exchanged with the trusted provider. For example, you can specify that the local attribute of any attribute in the Liberty profile (such as Informal Name) matches the remote attribute specified at the service provider.

3 Specify a set name, then click *Next*.

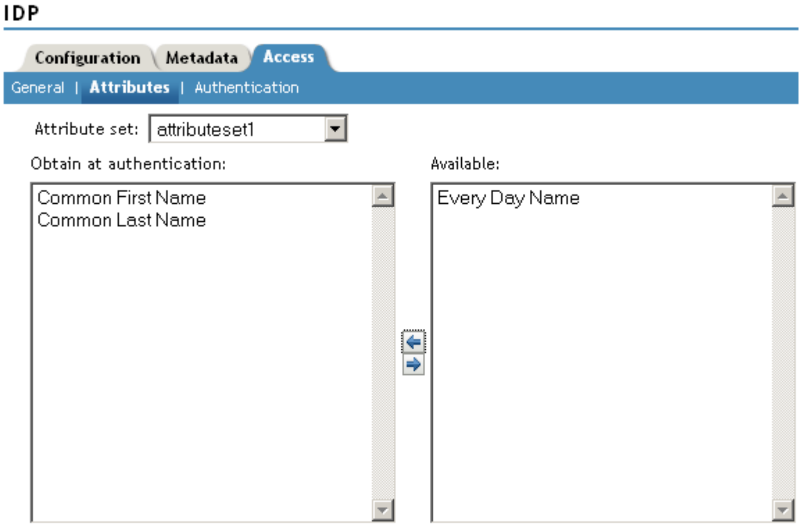
4 On the Define Attributes page, click *New*.

5 Select a local attribute.

6 Optionally, you can provide the name of the remote attribute.

7 Click *OK*, then click *Finish*.

After you select attributes, the system displays them on the Attributes page.



Attribute set: attributeset1

Obtain at authentication:

- Common First Name
- Common Last Name

Available:

- Every Day Name

You can select attributes from the *Available Attributes* field, and move them to the left side of the page. If you are an identity provider setting up a service provider, the left side of the page is used for attributes to be sent in an assertion to a service provider.

If you are a service provider setting up an identity provider, the attributes that you move to the left side of the page are those you want to be obtained by the service provider during authentication.

Configuring User Authentication and Federation

10

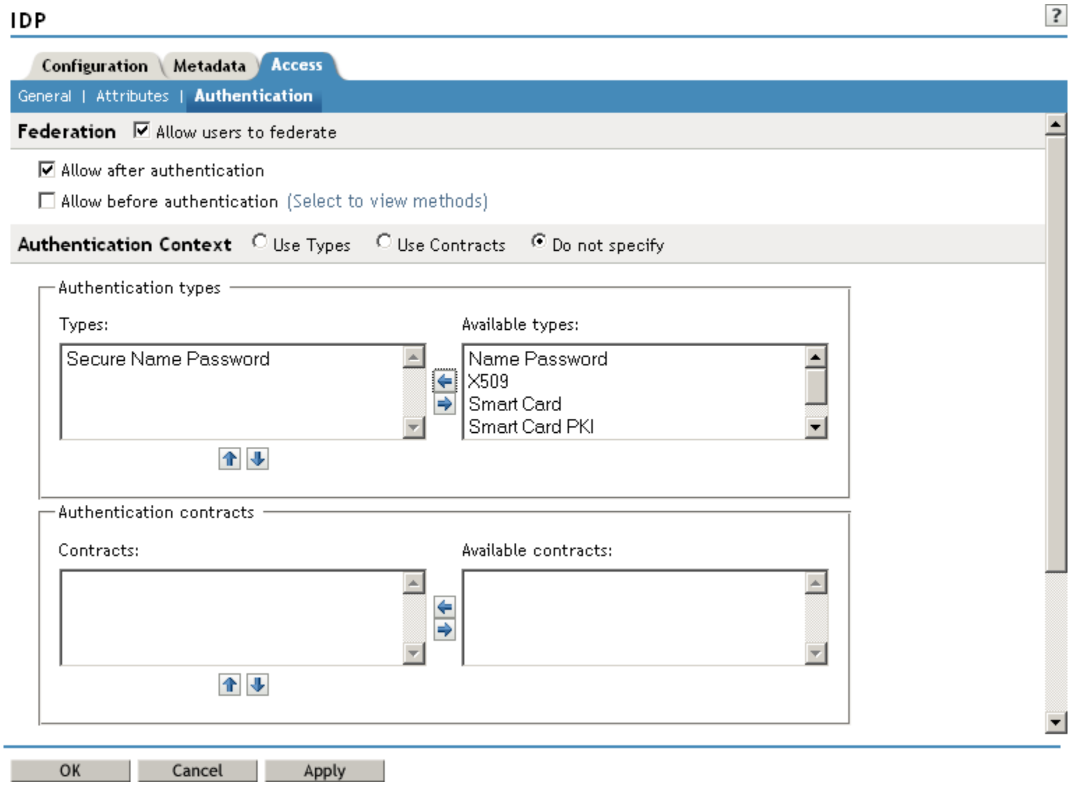
Configuring authentication involves determining how the trusted service provider interacts with the trusted identity provider during user authentication and federation. Examples include when authentication occurs and which authentication contracts to use. You can also configure the identification methods a service provider uses for provisioning unknown users.

10.1 Configuring Authentication for a Trusted Identity Provider

When users authenticate to a service provider, they can be given the option to federate their account identities with their preferred identity provider. This process creates an account association between the identity provider and service provider that enables single sign-on and single logout.

1 In the Administration Console, click *Access Manager Identity Servers > Servers > Edit > Liberty [or SAML 2.0] > [Identity Provider] > Access > Authentication*.

2 Click *Authentication*.



Federation ☑ Allow users to federate

☑ Allow after authentication

☐ Allow before authentication (Select to view methods)

Authentication Context ○ Use Types ○ Use Contracts ● Do not specify

Authentication types

Types: Secure Name Password

Available types:
Name Password
X509
Smart Card
Smart Card PKI

Authentication contracts

Contracts:

Available contracts:

3 Enable the following option:

Allow users to federate: Enables account federation. By enabling this option, you assume that a user account exists at the service provider and that the account can be associated with a user's account at the identity provider. If you do not use this feature, authentication is permitted but is not associated with a particular user account.

4 Specify when the federation request occurs:

Allow after authentication: Sends the federation request after the user has authenticated (logged in) to the service provider. When you set this option, users can federate from the Federations page in the Access Manager User Portal.

Allow before authentication: Specifies whether federation can occur when the user clicks the login link to the identity provider. Allowing federation in this method means that a user must be identified at a later time during the federation process. For this reason, when you click this option, the system displays additional options on the Authentication page, under *User Identification Methods*.

These options are discussed in Section 10.2, "Configuring User Identification Methods," on page 159.

5 Under *Authentication Context*, configure the following fields:

Use Types: Specifies whether to use authentication types. Select the types from the *Available types* field to specify which type to use for authentication between trusted service providers and identity providers. Standard types include Name/Password, X.509, Token, and so on.

Use Contracts: Specifies whether to use authentication contracts. Select the contract from the *Available contracts* list. For a contract to appear in the *Available contracts* list, the contract must have the *Satisfiable by External Provider* option enabled. To use the contract for federated authentication, the contract's URI must be the same on the identity provider and the service provider. For information about contract options, see Section 8.4, “Configuring Authentication Contracts,” on page 111.

Do not specify: Specifies that the identity provider can send any type of authentication to satisfy a service provider’s request, and instructs a service provider to not send a request for a specific authentication type or contract.

6. Under the *Options* heading, configure the following fields, as necessary:

   Response Protocol Binding: Select *Artifact* or *Post* or *None*. Artifact and Post are the two methods for transmitting assertions between the authenticating system and the target system.

   If you select *None*, you are letting the identity provider determine the protocol.

   Identity provider proxy redirects: Specifies whether or not the trusted identity provider can proxy the authentication request to another identity provider. A value of zero specifies that the trusted identity provider cannot redirect an authentication request. Values 1-5 determine the number of times the request can be proxied. Select *Configured on IDP* to let the trusted identity provider decide how many times the request can be proxied.

   Force authentication at the IDP: Specifies that the trusted identity provider must prompt the user for authentication, even if they are already logged in.

   Use automatic introduction: Automatically attempts single sign-on to this trusted identity provider.

7. Click *OK*.
8. On the Trusted Providers page, click *OK*.
9. Update the Identity Server configuration on the *Servers* page.

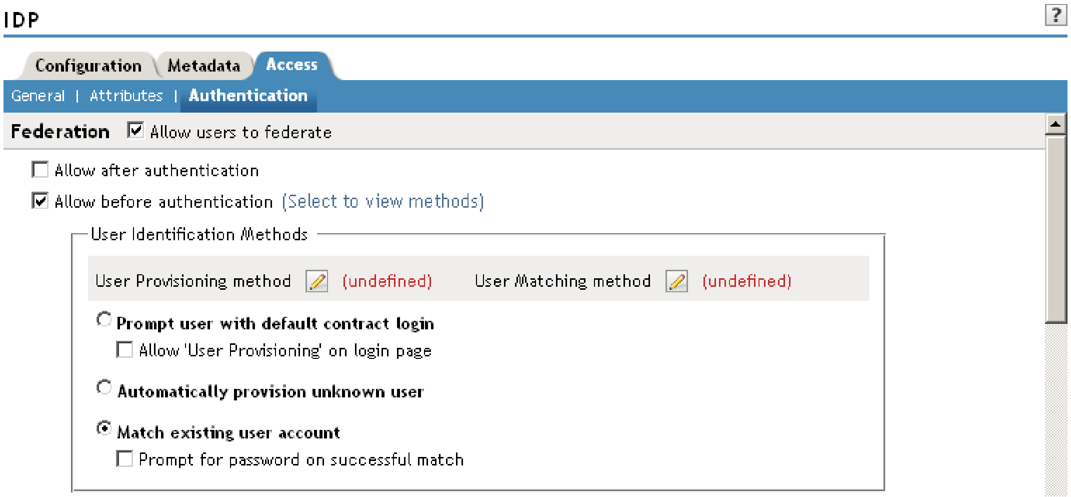
10.2 Configuring User Identification Methods

Three methods exist for you to identify users from a trusted identity provider. You can authenticate users by using the default authentication contract, match existing user accounts, or create new account with user provisioning. If there are problems during provisioning, you see error messages with more information.

- Section 10.2.1, “Selecting a User Identification Method,” on page 159
- Section 10.2.2, “Configuring the User Matching Method,” on page 160
- Section 10.2.3, “Defining the User Provisioning Method,” on page 161
- Section 10.2.4, “User Provisioning Error Messages,” on page 165

10.2.1 Selecting a User Identification Method

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty [or SAML 2.0] > [Identity Provider] > Access > Authentication*.
2. Enable *Allow before authentication*, then configure user provisioning or account matching as necessary as described below.



Federation ☑ Allow users to federate

☐ Allow after authentication

☑ Allow before authentication (Select to view methods)

User Identification Methods

User Provisioning method (undefined)

User Matching method (undefined)

○ Prompt user with default contract login

☐ Allow 'User Provisioning' on login page

○ Automatically provision unknown user

● Match existing user account

☐ Prompt for password on successful match

The system displays the following options on the Authentication page under *User Identification Methods*. These options are used to further configure how the service provider can authenticate an unrecognized user.

Authenticate user with default contract: Executes the default authentication contract.

- **Allow User Provisioning on login page:** Provides a button that the user can click to create an account.

  If you are a service provider using Active Directory, ensure that Active Directory is configured to use a secure port, such as 636, and that the user’s password conforms to the complexity policy. If you encounter a provisioning error, you must reset the password on the Windows* machine.

Automatically provision unknown users: Enables a service provider to trust unknown users that have authenticated to the trusted identity provider. User provisioning is used when no user account for federation exists at the service provider.

You must click *User Provisioning Method* to define user provisioning. See Section 10.2.3, “Defining the User Provisioning Method,” on page 161.

Match existing user accounts: Enables account matching. The service provider can uniquely identify a user in its directory by obtaining specific user attributes sent by the trusted identity provider.

You must click *User Matching Method* to define the match method. See Section 10.2.2, “Configuring the User Matching Method,” on page 160.

- **Prompt for password on successful match:** (Optional) Specifies whether to prompt the user for a password when the user’s name is matched to an account, to ensure that the account matches.

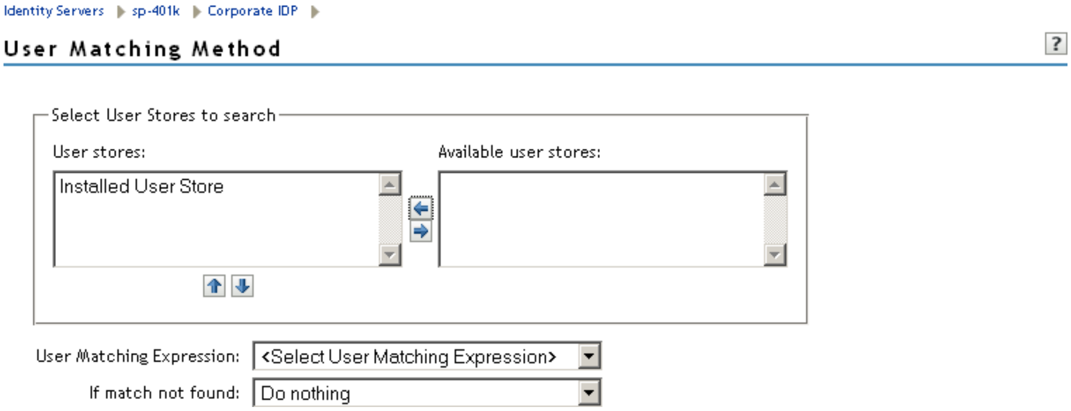
3. Click *OK*.
4. Click *OK* on the Trusted Providers page.
5. Click *Update Servers* on the Servers page.

10.2.2 Configuring the User Matching Method

If you enabled the *Match existing user account* option when selecting an identification method, you must configure the matching method.

Before you begin, enable the Liberty Personal Profile. See Section 12.2, “Enabling Web Services and Profiles,” on page 174.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty [or SAML 2.0]* > *[Identity Provider Name]* > *Access* > *Authentication*.
2. Click *Allow before authentication*.
3. Click *Match existing user account*.
4. Click *User Matching Method*.



Identity Servers ▸ sp-401k ▸ Corporate IDP ▸

User Matching Method

Select User Stores to search

User stores: Installed User Store

Available user stores:

User Matching Expression: <Select User Matching Expression>

If match not found: Do nothing

5. Select and arrange the user stores you want to use.
6. Set the matching expression as the default, or click *New* to create a look-up expression. See Section 7.3, “Configuring User Matching Expressions,” on page 85.
7. Specify what action to take if no match is found.

   You perform account matching before user provisioning, in order to prevent the creation of multiple accounts for one user. If no match is found, you can specify whether to:

   - Do nothing
   - Prompt the user for authentication
   - Automatically provision the user account
8. Click *Finish*.
9. On the Authentication page, click *OK*.
10. On the Trusted Providers page, click *OK*.
11. On the Servers page, click *Update Servers* to update the Identity Server configuration.

10.2.3 Defining the User Provisioning Method

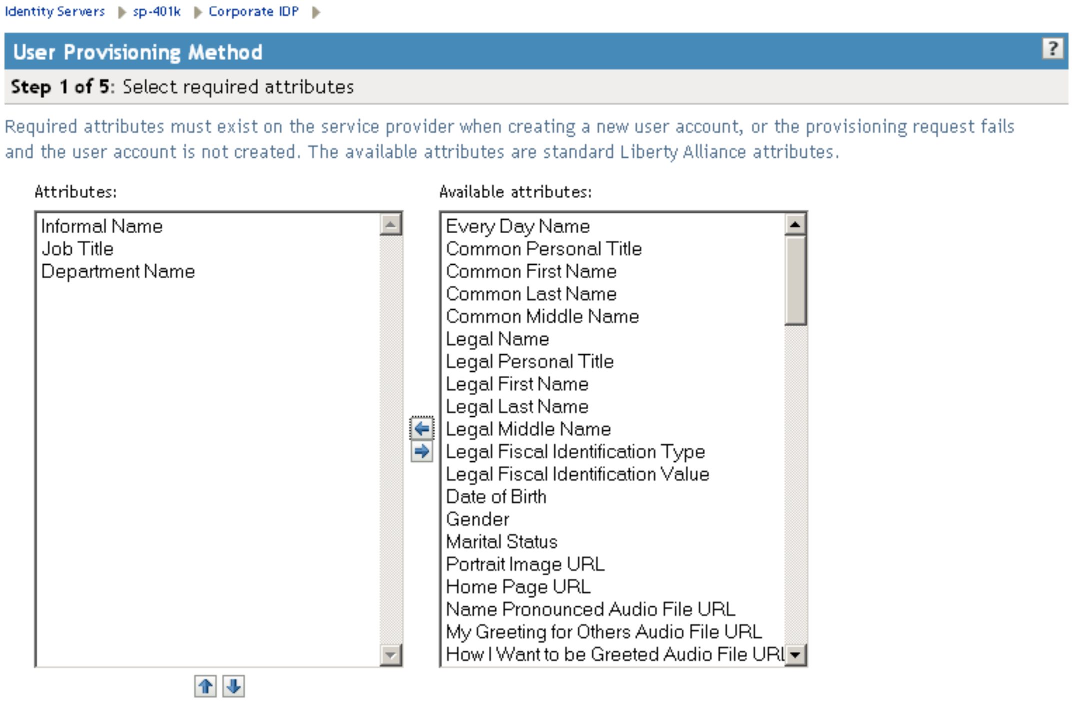
If you enabled *Automatically provision unknown users* when selecting an identification method, you must define the user provisioning method. This procedure involves selecting required and optional attributes that the service provider requests from the identity provider during provisioning.

Attribute Considerations

When a user object is created in the directory, some attributes are initially created with the value of NAM Generated. Afterwards, an attempt is made to write the required and optional attributes to the new user object. Because required and optional attributes are profile attributes, the system checks the write policy for the profile's Data Location Settings (specified in *Liberty* > *Web Service Provider*) and writes the attribute in either LDAP or the configuration store. In order for the LDAP write to succeed, each attribute must be properly mapped as an LDAP Attribute. Additionally, you must enable the read/write permissions for each attribute in the Liberty/LDAP attribute maps. See Section 12.9, "Mapping LDAP and Liberty Attributes," on page 184.

To configure user provisioning:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty [or SAML 2.0]* > *[Identity Provider]* > *Access* > *Authentication*.
2. Click *Allow before authentication*, then click *User Provisioning method*.



User Provisioning Method

Step 1 of 5: Select required attributes

Required attributes must exist on the service provider when creating a new user account, or the provisioning request fails and the user account is not created. The available attributes are standard Liberty Alliance attributes.

Attributes:

Informal Name
Job Title
Department Name

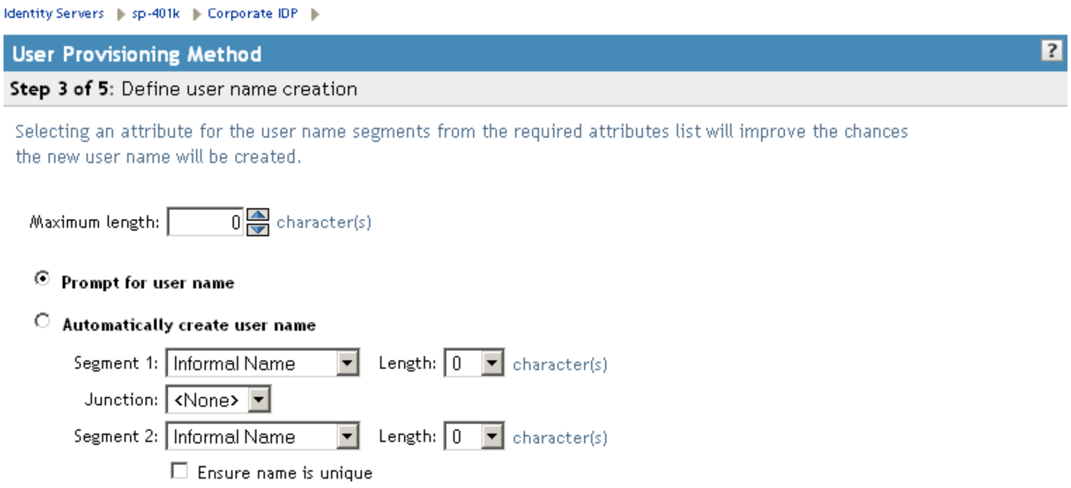
Available attributes:

Every Day Name
Common Personal Title
Common First Name
Common Last Name
Common Middle Name
Legal Name
Legal Personal Title
Legal First Name
Legal Last Name
Legal Middle Name
Legal Fiscal Identification Type
Legal Fiscal Identification Value
Date of Birth
Gender
Marital Status
Portrait Image URL
Home Page URL
Name Pronounced Audio File URL
My Greeting for Others Audio File URL
How I Want to be Greeted Audio File URL

3. Select the required attributes from the *Available Attributes* list and move them to the *Attributes* list.

   Required attributes are those used in the creation of a user name, or that are required when creating the account.
4. Click *Next*.
5. Select optional attributes from the *Available Attributes* list and move them to the *Attributes* list.

   This step is similar to selecting required attributes. However, the user provisioning request creates the user account whether or not optional attributes exist on the service provider.
6. Click *Next*.
7. Define how to create the username.



Identity Servers ▸ sp-401k ▸ Corporate IDP ▸

User Provisioning Method

Step 3 of 5: Define user name creation

Selecting an attribute for the user name segments from the required attributes list will improve the chances the new user name will be created.

Maximum length: 0 character(s)

● **Prompt for user name**

○ **Automatically create user name**

Segment 1: Informal Name Length: 0 character(s)

Junction: <None>

Segment 2: Informal Name Length: 0 character(s)

☐ Ensure name is unique

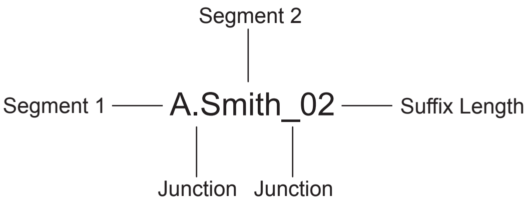
You can specify whether users are prompted to create their own usernames or whether the system automatically creates usernames. Selecting an attribute for the username segments from the required attributes list improves the chances that a new username is successfully created.

Maximum length: The maximum length of the user name. This value must be between 1 and 50.

Prompt for user name: Enables users to create their own usernames.

Automatically create user name: Specifies that the system creates usernames. You can configure the segments for the system to use when creating usernames and configure how the names are displayed.

For example, if you are using the required attributes of Common First Name and Common Last Name, a username for Adam Smith might be generated as A.Smith_02, as shown in the following illustration:



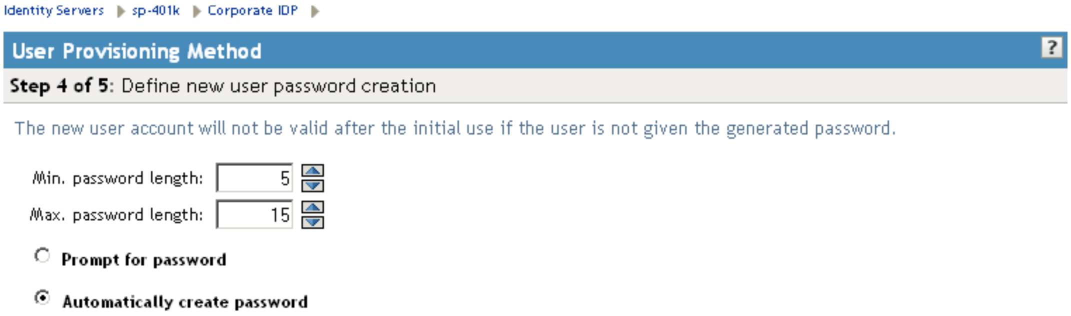
Use the following settings to specify how this is accomplished:

- **Segment 1:** The required attribute to use as the first segment for the user name. The values displayed in this drop-down menu correspond to the required attributes you selected. For example, you might select Common First Name to use for *Segment 1*.
- **Length:** The length of the first attribute segment. For example, if you selected Common First Name for the *Segment 1* value, setting the length to 1 specifies that the system uses the first letter of the Common First Name attribute. Therefore, Adam Smith would be ASmith.
- **Junction:** The type of junction to use between the attributes of the user name, such as no space, or a hyphen, or a period. Adam Smith would display as A.Smith.
- **Segment 2:** The required attribute to use as the second segment for the user name. The values displayed in this drop-down menu correspond to the required attributes you selected. For example, you might select Common Last Name to use for *Segment 2*.

- **Length:** The length of the second attribute segment. For example, if you selected Common Last Name for the *Segment 2* value, you might set the length to *All*, so that the full last name is displayed. However, the system does not allow more than 20 characters for the length of segment 2.
- **Ensure name is unique:** Applies a suffix to the colliding name until a unique name is found, if using attributes causes a collision with an existing name. If no attributes are provided, or the lengths for them are 0, and this option is selected, the system creates a unique name.

8 Click *Next*.

9 Specify password settings.



Identity Servers ▸ sp-401k ▸ Corporate IDP ▸

User Provisioning Method

Step 4 of 5: Define new user password creation

The new user account will not be valid after the initial use if the user is not given the generated password.

Min. password length: 5

Max. password length: 15

○ Prompt for password

● Automatically create password

Use this page to specify whether to prompt the user for a password or to create a password automatically.

Min. password length: The minimum length of the password.

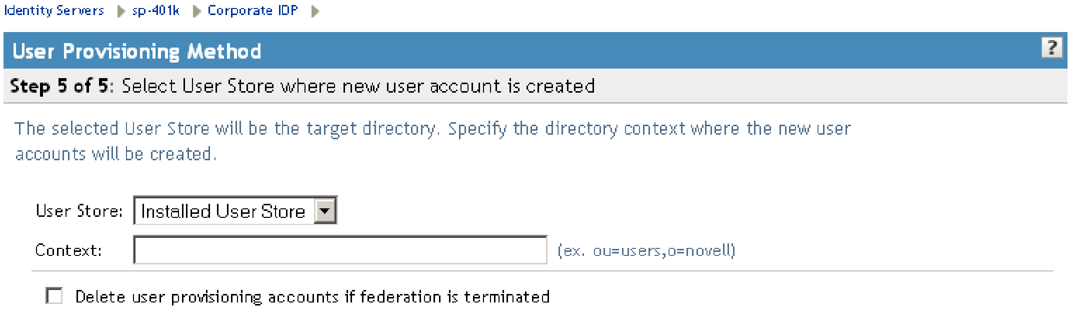
Max. password length: The maximum length of the password.

Prompt for password: Prompts the user for a password.

Automatically create password: Specifies whether to automatically create passwords.

10 Click *Next*.

11 Specify the user store and context in which to create the account.



Identity Servers ▸ sp-401k ▸ Corporate IDP ▸

User Provisioning Method

Step 5 of 5: Select User Store where new user account is created

The selected User Store will be the target directory. Specify the directory context where the new user accounts will be created.

User Store: Installed User Store

Context: (ex. ou=users,o=novell)

☐ Delete user provisioning accounts if federation is terminated

User Store: The user store in which to create the new user account.

Context: The context in the user store you want accounts created.

The system creates the user within a specific context; however, uniqueness is not guaranteed across the directory.

Delete user provisioning accounts if federation is terminated: Specifies whether to automatically delete the provisioned user account at the service provider if the user terminates his or her federation between the identity provider and service provider.

12 Click *Finish*.

13 On the Authentication page, click *OK*.

14 On the Trusted Providers page, click *OK*.

15 On the Servers page, click *Update Servers* to update the Identity Server configuration.

10.2.4 User Provisioning Error Messages

The following error messages are displayed for the end user if there are problems during provisioning.

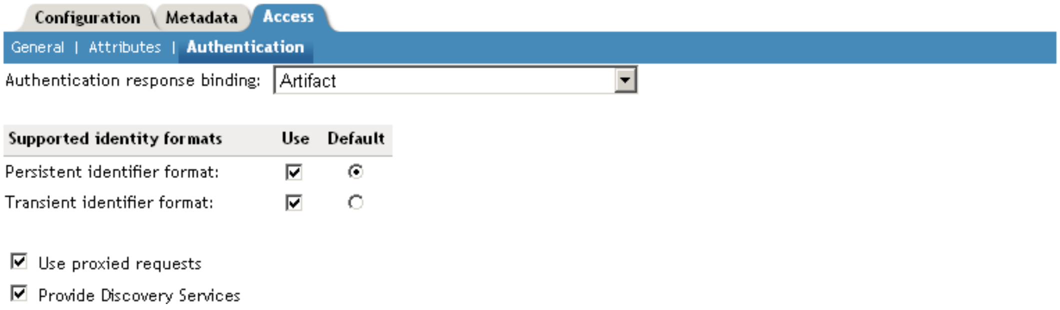
Table 10-1 *Provisioning Error Messages*

Error Message	Cause
`Username length cannot exceed (?) characters.`	The user entered more characters for a user name than is allowed, as specified by the administrator.
`Username is not available.`	The user entered a name that already exists in the directory.
`Passwords don’t match.`	The user provided two password values that do not match.
`Passwords must be between (x) and (y) characters in length.`	The user provided password values that are either too short or too long.
`Username unavailable.`	The provisioned user account was deleted without first defederating the user. Remove orphaned identity objects from the configuration datastore. **IMPORTANT:** Only experienced LDAP users should remove orphaned identity objects from the configuration datastore. You must ensure that the objects you are removing are orphaned. Otherwise, you create orphaned objects by mistake.
`Unable to complete authentication request.`	Can occur when users are allowed to create accounts from a service provider’s login page, when the service provider uses Active Directory for the user store. The password provided does not conform to the Windows password complexity policy in Active Directory. Ensure that Active Directory is configured to use a secure port, such as 636, and that the user’s password conforms to the complexity policy. If you encounter this error, you must reset the password on the Windows* machine.

10.3 Configuring Authentication for a Trusted Service Provider

After you create a trusted service provider, you can configure how your Identity Server responds to authentication requests from the service provider.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty* > *[Service Provider]* > *Access* > *Authentication*.



Authentication response binding: Artifact

Supported identity formats	Use	Default
Persistent identifier format:	☑	●
Transient identifier format:	☑	○

☑ Use proxied requests

☑ Provide Discovery Services

2 Fill in the following fields as required:

Authentication Response Binding: Specifies whether to use *Artifact* or *Post* if the request from the trusted service provider does not specify a response binding. Select *Artifact* to provided an increased level of security by using a back-channel means of communication between the two servers. Select *Post* to use HTTP redirection to accomplish communication between servers.

- **Persistent Identifier Format:** Specifies whether to use this format and make it the default identifier format. A persistent identifier is written to the directory and remains intact between sessions.
- **Transient Identifier Format:** Specifies whether to use this format and make it the default identifier format. A transient identifier expires between sessions.

Use Proxied Requests: Enables proxying for the service provider. If disabled, no proxying is allowed.

For example, the service provider can authenticate a user to IDP B through IDP A, when no trust relationship exists between the service provider and IDP B. This feature is allowed by default. However, you can disable the service provider’s ability to use proxied requests. In order to use this, you must specify Silent Login on IDP A.

Proxying can also be used to achieve single sign-on when the trust authentication types and contracts differ between identity providers, or when identity providers are using multiple protocols, such as when one identity provider communicates via SAML 2.0, and another uses Liberty.

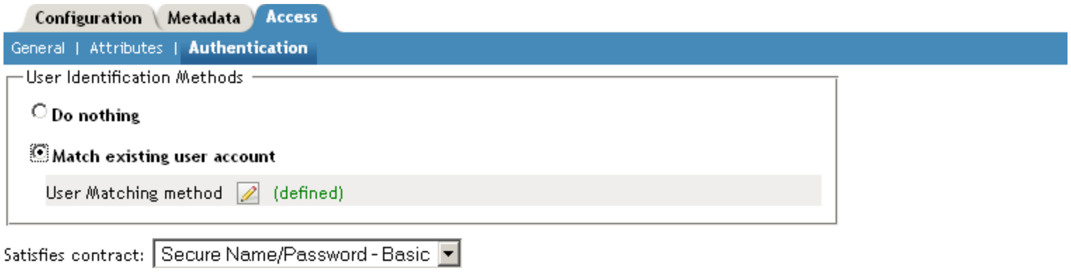
Provide Discovery Services: Advertises to the service provider the Web services available at the Identity Server. This option is required if the identity provider is to provide services to the service provider.

3 Click *OK*.

10.4 Configuring User Identification Methods for SAML 1.1 Trusted Identity Providers

Two methods exist for identifying users from a trusted identity provider. You can specify that no account matching needs to occur, or you can configure a match method. You configure a match method when you want to use attributes from this trusted identity provider to uniquely identify a user.

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > SAML 1.1 > [Identity Provider] > Access > Authentication.*


Configuration
Metadata
Access
General | Attributes | Authentication
User Identification Methods

○ Do nothing

● Match existing user account

User Matching method (defined)

Satisfies contract: Secure Name/Password - Basic

2. Configure the following options as necessary:

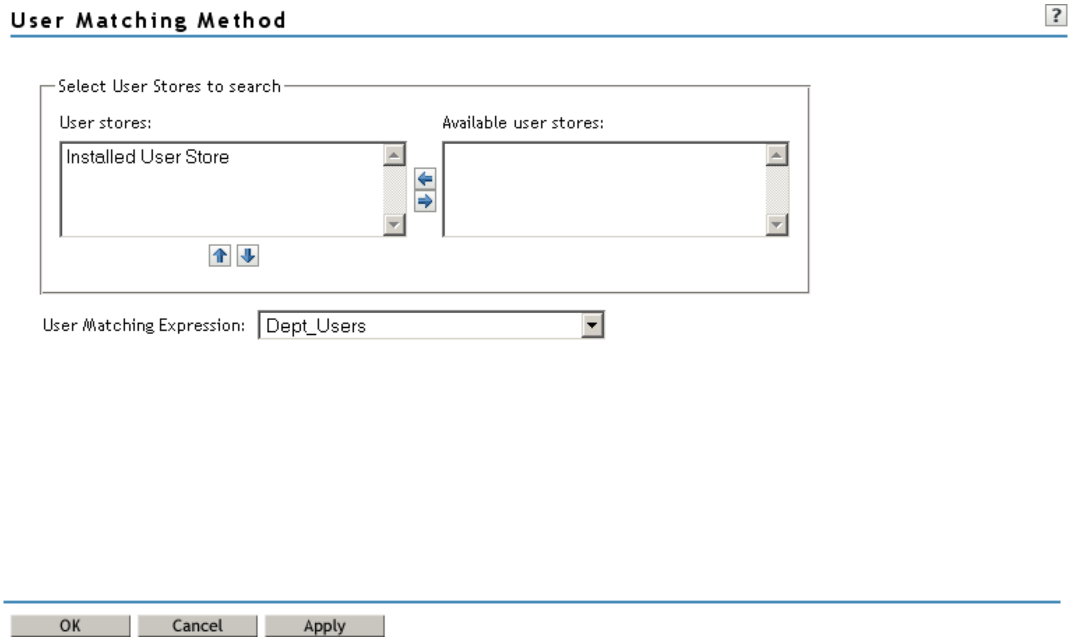
   Do nothing: Specifies that the service provider does not match user accounts. This option allows you to authenticate the session without identifying a user account.

   Match existing user accounts: Authenticates a user by matching a user account. This option requires that you set up the match method. (See Step 3.)

   Satisfies contract: The contract that is satisfied by the assertion received from the identity provider. Because SAML 1.1 does not use contracts and because the Identity Server is contract-based, this setting permits an association to be made between a contract and a SAML 1.1 assertion.

   Use caution when assigning the contract to associate with the assertion, because it is possible to imply that authentication has occurred, when it has not. For example, if a contract is assigned to the assertion, and the contract has two authentication methods (such as one for name/password and another for X.509), the server sending the assertion might use only name/password, but the service provider might assume that X.509 authentication took place and then incorrectly assert it to another server.

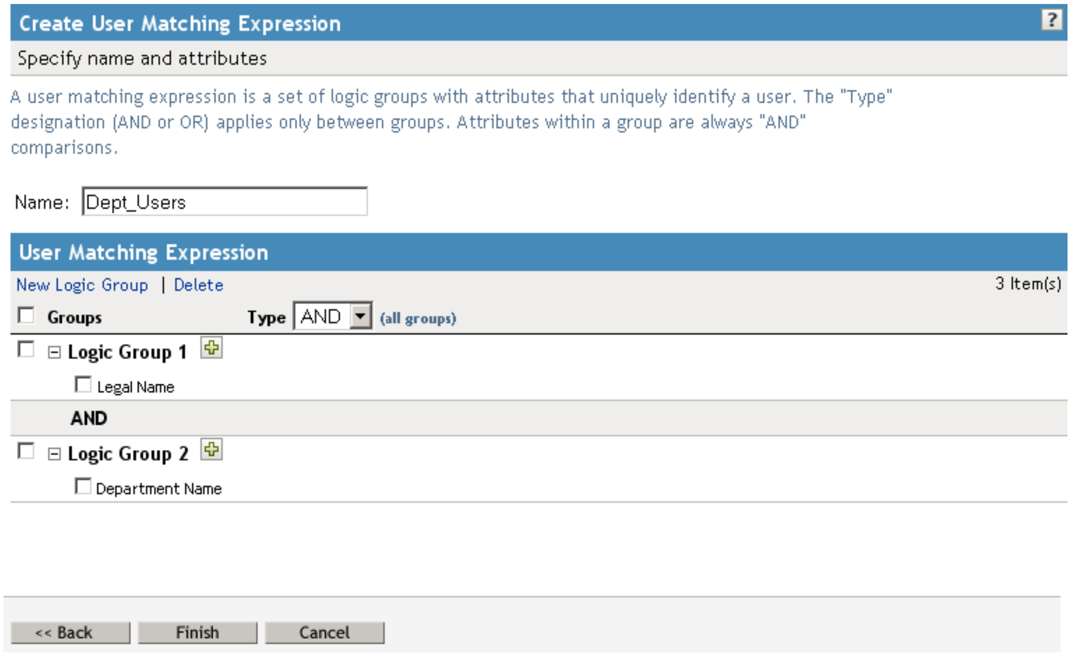
3. To configure the match method, click *User Matching Method*.



4 To configure user matching, fill in the following fields:

Select User Stores to search: Select and order the user stores you want to use in the search.

User Matching Expression: Set the matching expression as the default, or click *New* to create a look-up expression.



A user matching expression is a set of logic groups with attributes that uniquely identify a user. The "Type" designation (AND or OR) applies only between groups. Attributes within a group are always "AND" comparisons.

Name: Dept_Users

A user matching expression is a set of logic groups with attributes that uniquely identify a user. User matching expressions enable you to map the Liberty attributes to the correct LDAP attributes during searches. You must know the LDAP attributes that are used to name the users in the user store and create the user's distinguished name.

In order to use user matching, you must enable the Personal Profile on the identity provider and the service provider. See Section 12.2, “Enabling Web Services and Profiles,” on page 174.

5. Click *Finish*.
6. Select the new expression on the User Method Matching page, then click *OK*.
7. Click *OK* on the Authentication page, then click *OK* on the Trusted Providers page.
8. Update the Identity Server configuration on the Servers page, as prompted.

10.5 Specifying a SAML Audience URI

When an identity provider sends an assertion to a service provider, the assertion can be restricted to an intended audience. The intended audience is defined to be any abstract URI in SAML 1.1. The URI reference can also identify a document that describes the terms and conditions of audience membership.

In the Liberty specification, which uses SAML assertions, the audience is the provider ID. When you first set up a SAML partnership, adding audience restrictions conditions can add unnecessary complexity.

1. In the Administration Console, click *Access Manager > Identity Servers > [Configuration Assignment] > SAML 1.1 > [Service Provider] > Access Audiences*.
2. Click *New*.
3. Specify the *SAML Audience URI* value, then click *OK*.

Configuring Communication Profiles

11

You can configure the methods of communication that are available at the server for requests and responses sent between providers. These settings affect the metadata for the server and should be determined prior to publishing to other sites.

In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > [Protocol] > Profiles*.

Artifact Resolution: (SAML 2.0 only) The assertion consumer service at the service provider performs a back-channel exchange with the artifact resolution service at the identity provider. Artifacts are small data objects pointing to larger SAML protocol messages. They are designed to be embedded in URLs and conveyed in HTTP messages.

Login: Specifies whether to support Artifact or Post binding for login. The Artifact binding provides an increased level of security by using a back-channel means of communication between the two servers during authentication. The Post method uses HTTP redirection to accomplish communication between servers.

Single Logout: Enables the identity provider or service provider to accept HTTP and SOAP requests. Typically, you select both of these options. SOAP is used if both options are selected, or if the service provider has not specified a preference.

HTTP Redirect: A browser-based method that uses HTTP 302 redirects or HTTP GET requests to communicate requests from this identity site to the service provider. SAML messages are transmitted within URL parameters.

Federation Termination: (Liberty only) Specifies whether to use HTTP or SOAP profiles. Typically, you select both of these options, which enables the identity provider or service provider to accept both HTTP and SOAP requests. SOAP is the default setting if the service provider has not specified a preference.

Register Name: (Liberty only) Specifies whether to use HTTP or SOAP profiles. Typically, you select both of these options, which enables the identity provider or service provider to accept both HTTP and SOAP requests. SOAP is the default setting if the service provider has not specified a preference.

Name Management: (SAML 2.0 only) Specifies the binding protocol for the SAML Name Identifier Management profile. Name management is how the system manages the sharing of common identifiers for a principal between identity and service providers. When an identity provider has exchanged a persistent identifier for the principal with a service provider, the providers share the common identifier for a length of time. When either the identity or service provider changes the format or value to identify the principal, the system can ensure that the new format or value is properly transmitted.

Configuring Liberty Web Services

12

A Web service uses Internet protocols to provide a service. It is an XML-based protocol transported over SOAP, or a service whose instances and data objects are addressable via URIs.

Access Manager consists of several elements that comprise Web services:

- **Web Service Framework:** Manages all Web services. The framework defines SOAP header blocks and processing rules that enable identity services to be invoked via SOAP requests and responses.
- **Web Service Provider:** An entity that provides data via a Web service. In Access Manager, Web service providers host Web service profiles, such as the Employee Profile, Credential Profile, Personal Profile, and so on.
- **Web Service Consumer:** An entity that uses a Web service to access data. Web service consumers discover resources at the Web service provider, and then retrieve or update information about a user, or on behalf of a user. Resource discovery among trusted partners is necessary because a user might have many kinds of identities (employee, spouse, parent, member of a group), as well as several identity providers (employers or other commercial Web sites).
- **Discovery Service:** The service assigned to an identity provider that enables a Web Service Consumer to determine which Web service provider provides the required resource.
- **LDAP Attribute Mapping:** Access Manager's solution for mapping Liberty attributes with established LDAP attributes.

This section describes the following topics:

- Section 12.1, "Configuring the Web Services Framework," on page 174
- Section 12.2, "Enabling Web Services and Profiles," on page 174
- Section 12.3, "Editing Web Service Descriptions," on page 175
- Section 12.4, "Configuring Credential Profile Security and Display Settings," on page 176
- Section 12.5, "Configuring Service and Profile Details," on page 178
- Section 12.6, "Customizing Attribute Names," on page 180
- Section 12.7, "Editing Web Service Policies," on page 181
- Section 12.8, "Configuring the Web Service Consumer," on page 184
- Section 12.9, "Mapping LDAP and Liberty Attributes," on page 184

For additional resources about the Liberty Alliance specifications, visit the Liberty Alliance Specification (http://www.projectliberty.org/resources/specifications.php) page.

12.1 Configuring the Web Services Framework

The Web Services Framework page lets you edit and manage all the details that pertain to all Web services. This includes the framework for building interoperable identity services, permission-based attribute sharing, identity service description and discovery, and the associated security mechanisms.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Framework*.

2 Fill in the following fields:

   Enable Framework: Enables Web Services Framework.

   Axis SOAP Engine Settings: Axis is the SOAP engine that handles all Web service requests and responses. Web services are deployed using XML-based files known as Web service deployment descriptors (WSDD). On startup, Access Manager automatically creates the server-side and client-side configuration for Axis to handle all enabled Web services. If you need to override this default configuration, use the *Axis Server Configuration WSDD XML* field and the *Axis Client Configuration WSDD XML* field to enter valid WSDD XML. If either or both of these controls contain valid XML, then Access Manager does not automatically create the configuration (server or client) on startup.

3 Click *OK*.

12.2 Enabling Web Services and Profiles

After a service has been discovered and authorization data has been received from a trusted identity provider, the Web service consumer can invoke the service at the Web service provider. A Web service provider is the hosting or relying entity on the server side that can make access control decisions based on this authorization data and upon its business practices and preferences.

1 In the Administration Console click *Identity Servers > Servers > Edit > Liberty > Web Service Providers*.

2 Select one of the following services:

   Authentication Profile: Allows the system to access the roles and authentication contracts in use by current authentications. This profile is enabled by default so that embedded service providers can evaluate roles in policies. This profile can be disabled. When disabled, all devices assigned to use this Identity Server cluster configuration cannot determine which roles a user has been assigned, and the devices evaluate policies as if the user has no roles.

   WARNING: Do not delete this profile. In normal circumstances, this profile is used only by the system.

   Credential Profile: Allows users to define information to keep secret. It uses encryption to store the data in the directory the user profile resides in.

   Custom Profile: Used to create custom attributes for general use.

   Discovery: Allows requesters to discover where the resources they need are located. Entities can place resource offerings in a discovery resource, allowing other entities to discover them. Resources might be a user's credit card information, a personal profile, calendar, travel preferences, and so on.

   Employee Profile: Allows you to manage employment-related information and how the information is shared with others. A company address book that provides names, phones, office locations, and so on, is an example of an employee profile.

LDAP Profile: Allows you to use LDAP attributes for authorization and general use.

Personal Profile: Allows you to manage personal information and to determine how to share that information with others. A shopping portal that manages the user's account number is an example of a personal profile.

User Interaction: Allows you to set up a trusted user interaction service, used for identity services that must interact with the resource owner to get information or permission to share data with another Web service consumer. This profile enables a Web service consumer and Web service provider to cooperate in redirecting the resource owner to the Web service provider and back to the Web service consumer.

3. Click *Enable*, then click *OK*.
4. On the Servers page, click *Update Servers* to update the Identity Server configuration.

12.3 Editing Web Service Descriptions

All of the Description pages on each profile are identical. You can define how a service provider gains access to portions of the user's identity information that can be distributed across multiple providers. The service provider uses the Discovery Service to ascertain the location of a specific identity service for a user. The Discovery Service enables various entities to dynamically and securely discover a user's identity service, and it responds, on a permission basis, with a service description of the desired identity service.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty* > *Web Service Provider*.
2. Click the profile or service.
3. Click *Descriptions*.
4. Click the description name, or click *New*.
5. Fill in the following fields:

   Name: The Web Service Description name.

   Security Mechanism: (Required) Liberty uses channel security (TLS 1.0) and message security in conjunction with the security mechanism. Channel security addresses how communication between identity providers, service providers, and user agents is protected. For authentication, service providers are required to authenticate identity providers by using identity provider server-side certificates. Identity providers have the option to require authentication of service providers by using service provider client-side certificates.

   Message security addresses security mechanisms applied to the discrete Liberty protocol messages passed between identity providers, service providers, and user agents.

   Select the mechanism for message security. Message authentication mechanisms indicate which profile is used to ensure the authenticity of a message.

   - **X.509:** Used for message exchanges that generally rely upon message authentication as the principle factor in making authorization decisions.
   - **SAML:** Used for message exchanges that generally rely upon message authentication as well as the conveyance and attestation of authorization information.
   - **Bearer:** Based on the presence of the security header of a message. In this case, the bearer token is verified for authenticity rather than proving the authenticity of the message.

6. Under Select Service Access Method, click either *Brief Service Access Method* or *WSDL Service Access Method*.

Brief Service Access Method: Provides the information necessary to invoke basic SOAP-over-HTTP-based service instances without using WSDL.

- **EndPoint URL:** This is the SOAP endpoint location at the service provider to which Liberty SOAP messages are sent. An example of this for the Employee Profile is [BASEURL]/services/IDSISEmployeeProfile. If the service instance exposes an endpoint that is different from the logically generated concrete WSDL, you must use the WSDL URI instead.

  A WSF service description endpoint cannot contain double-byte characters.
- **SOAP Action:** The SOAP action HTTP header required on HTTP-bound SOAP messages. This header can be used to indicate the intent of a SOAP message to the recipient.

WSDL Service Access Method: Specify the method used to access the WSDL service. WSDL (Web Service Description Language) describes the interface of a Web service.

- **Service Name Reference:** A reference name for the service.
- **WSDL URI:** Provides a URI to an external concrete WSDL resource containing the service description. URIs need to be constant across all implementations of a service to enable interoperability.

7. Click *OK*.
8. Update the Identity Server configuration.

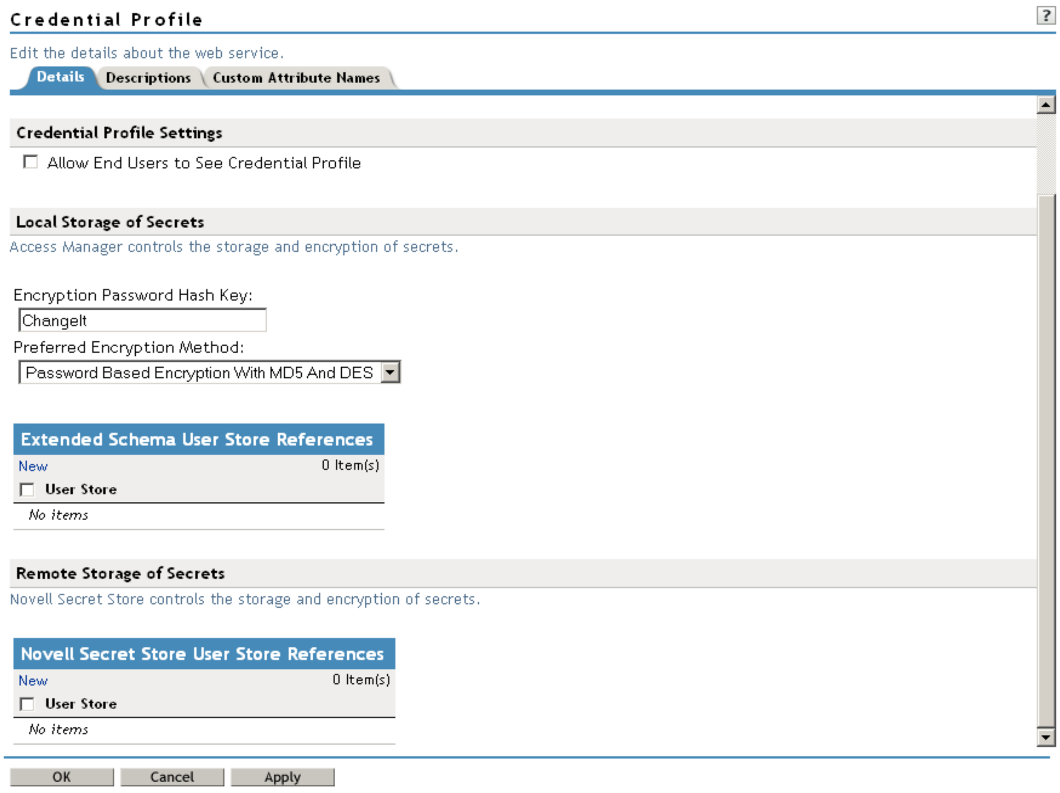
12.4 Configuring Credential Profile Security and Display Settings

On the Credential Profile Details page, you can specify whether this profile is displayed for end users, and determine how you control and store encrypted secrets. You can store and access secrets locally or on remote eDirectory™ servers that are running Novell® SecretStore®. For general information about this product, see the *Novell SecretStore Administration Guide* (http://www.novell.com/documentation/secretstore33/pdfdoc/nssadm/nssadm.pdf).

For information about creating shared secrets for Form Fill and Identity Injection policies, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.

To configure the Credential Profile:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Servers* > *Edit* > *Liberty* > *Web Service Providers*.
2. Click *Credential Profile*.



Edit the details about the web service.

Credential Profile Settings

☐ Allow End Users to See Credential Profile

Local Storage of Secrets

Access Manager controls the storage and encryption of secrets.

Encryption Password Hash Key: ChangeIt

Preferred Encryption Method: Password Based Encryption With MD5 And DES

Extended Schema User Store References

☐ User Store

No items

Remote Storage of Secrets

Novell Secret Store controls the storage and encryption of secrets.

Novell Secret Store User Store References

☐ User Store

No items

3 On the Credential Profile Details page, fill in the following fields as necessary:

Display name: The name you want to display for the Web service.

Have Discovery Encrypt This Service's Resource Ids: Specifies whether the Discovery Service encrypts resource IDs. A resource ID is an identifier used by Web services to identify a user. The Discovery Service returns a list of resource IDs when a trusted service provider queries for the services owned by a given user. The Discovery Service has the option of encrypting the resource ID or sending it unencrypted. Encrypting resource IDs is disabled by default.

4 Under *Credential Profile Settings*, enable the following option if necessary:

Allow End Users to See Credential Profile: Specifies whether to display or hide the Credential Profile in the Access Manager User Portal. Profiles are viewed on the My Profile page, where the user can modify his or her profile.

5 Specify how you want to control and store secrets:

5a To locally control and store secrets, configure the following fields:

Encryption Password Hash Key: (Required) Specifies the password that you want to use as a seed to create the encryption algorithm. To increase the security of the secrets, we recommend that you change the default password to a unique alphanumeric value.

Preferred Encryption Method: Specify the preferred encryption method. Select the method that complies with your security model:

- **Password Based Encryption With MD5 and DES:** MD5 is an algorithm that is used to verify data integrity. Data Encryption Standard (DES) is a widely used method of data encryption using a private key.

- **DES:** Data Encryption Standard (DES) is a widely used method of data encryption using a private key. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.
- **Triple DES:** A variant of DES in which data is encrypted three times with standard DES using two different keys.

5b Specify where to store secret data. (For more information about setting up a user store for secret store, see Section 8.1.4, “Configuring a User Store for Secrets,” on page 94.)

- To have the secrets stored in the configuration database, do not configure the list in the *Extended Schema User Store References* section. You only need to configure the fields in Step 5a.
- To store the secrets in your LDAP user store, click *New* in *Extended Schema User Store References* and configure the following fields:

  User Store: Select a user store where secret data is stored.

  Attribute Name: Specify the LDAP attribute of the User object that can be used to store the secrets. When a user authenticates using the user store specified here, the secret data is stored in an XML document of the specified attribute of the user object. This attribute should be a single-valued case ignore string that you have defined and assigned to the user object in the schema
- To use Novell SecretStore to remotely store secrets, click *New* under *Novell Secret Store User Store References.*

  Click the user store that you have configured for SecretStore.

  Secure LDAP must be enabled between the user store and the Identity Server in order to add this user store reference.

5c Click *OK*.

6 Click *OK*.

7 On the Identity Server page, update the Identity Server.

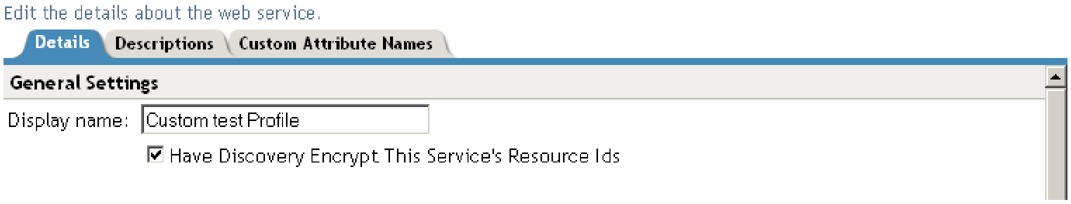
12.5 Configuring Service and Profile Details

The settings on the Details page are identical for the Employee, Custom, and Personal Profiles. This page allows you to specify the display name, resource ID encryption, and how the system reads and writes data.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Provider*.

2 Click either *Custom Profile*, *Employee Profile*, or *Personal Profile*, depending on which profile you want to edit.

3 Click the *Details* tab (it is displayed by default).



Edit the details about the web service.

General Settings

Display name: Custom test Profile

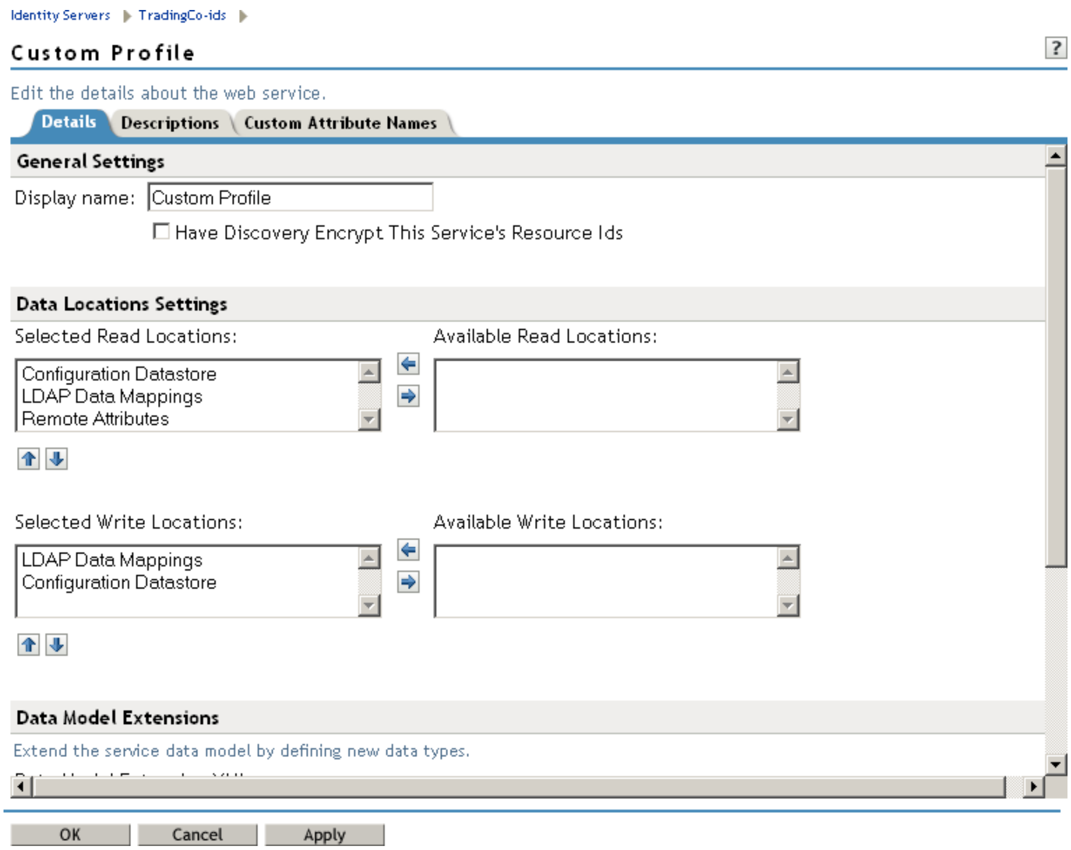
☑ Have Discovery Encrypt This Service's Resource Ids

4 Specify the general settings, as necessary:

Display Name: The Web service name. This specifies how the profile is displayed in the Administration Console.

Have Discovery Encrypt This Service's Resource Ids: Specifies whether the Discovery Service encrypts resource IDs. A resource ID is an identifier used by Web services to identify a user. The Discovery Service returns a list of resource IDs when a trusted service provider queries for the services owned by a given user. The Discovery Service has the option of encrypting the resource ID or sending it unencrypted.

5. Specify data location settings:



Custom Profile

Edit the details about the web service.

General Settings

Display name: Custom Profile

☐ Have Discovery Encrypt This Service's Resource Ids

Data Locations Settings

Selected Read Locations:
Configuration Datastore
LDAP Data Mappings
Remote Attributes

Available Read Locations:

Selected Write Locations:
LDAP Data Mappings
Configuration Datastore

Available Write Locations:

Data Model Extensions

Extend the service data model by defining new data types.

The following settings apply only to the Custom, Employee, and Personal Profiles.

Selected Read Locations: The list of selected locations from which the system reads attributes containing profile data. If you add multiple entries to this list, the system searches attributes in each location in the order you specify. When a match is found for an attribute, the other locations are not searched. Use the Up/Down and Left/Right arrows to control which locations are selected and the order in which to read them. Read locations can include:

- **Configuration Datastore:** Liberty attribute values can be stored in the configuration store of the Administration Console. If your users have access to the User Portal, they can add values to a number of Liberty attributes.
- **LDAP Data Mappings:** If you have mapped a Liberty attribute to an LDAP attribute in your user store, the values can be read from the LDAP user store. To create LDAP attribute maps, see Section 12.9, "Mapping LDAP and Liberty Attributes," on page 184.
- **Remote Attributes:** If you set up federation, the Identity Server can read attributes from these remote service providers. Sometimes, the service provider is set up to push at set of

attribute values when the user logs in. These pushed attributes are cached, and the Identity Server can quickly read them. If a requested attribute has not been pushed, a request for the Liberty attribute is sent to remote service provider. This can be time consuming, especially if the user has federated with more than one remote service provider. *Remote Attributes* should always be the last item in this list.

Available Read Locations: The list of available locations from which the system can read attributes containing profile data. Any location in this list is currently not being used.

Selected Write Locations: The list of selected locations to write attribute data to. If you add multiple entries to this list, the system searches attributes in each location in the order you specify. When a match is found for an attribute, the other locations are not searched. Use the Up/Down and Left/Right arrows to control which locations are selected and the order in which they are selected.

- **Configuration Datastore:** Liberty attribute values can be stored in the configuration store of the Administration Console. The Identity Server can write values to these attributes. If this location appears first in the list of *Selected Write Locations*, all Liberty attribute values are written to this location. If you want values written to the LDAP user store, the *LDAP Data Mappings* location must appear first in the list.
- **LDAP Data Mappings:** If you have mapped a Liberty attribute to an LDAP attribute in your user store, the Identity Server can write values to the attribute in the LDAP user store. To create LDAP attribute maps, see Section 12.9, "Mapping LDAP and Liberty Attributes," on page 184.

Available Write Locations: The list of available locations to write attributes containing profile data. Any location in this list is currently not being used.

6 (Optional) Specify data model extensions.

Data Model Extension XML: The data model for some Web services is extensible. You can enter XML definitions of data model extensions in this field. Data model extensions hook into the existing Web service data model at predefined locations.

All schema model extensions reside inside of a schema model extension group. The group exists to bind model data items together under a single localized group name and description. Schema model extension groups can reside inside of a schema model extension root or inside of a schema model extension. There can only be one group per root or extension. Each root is hooked into the existing Web service data model. Multiple roots can be hooked into the same location in the existing Web service data model. This conceptual model applies to the structure of the XML that is required to define data model extensions.

See Appendix D, "Data Model Extension XML," on page 707 for more information.

7 Click *OK*, then click *OK* on the Web Service Provider page.

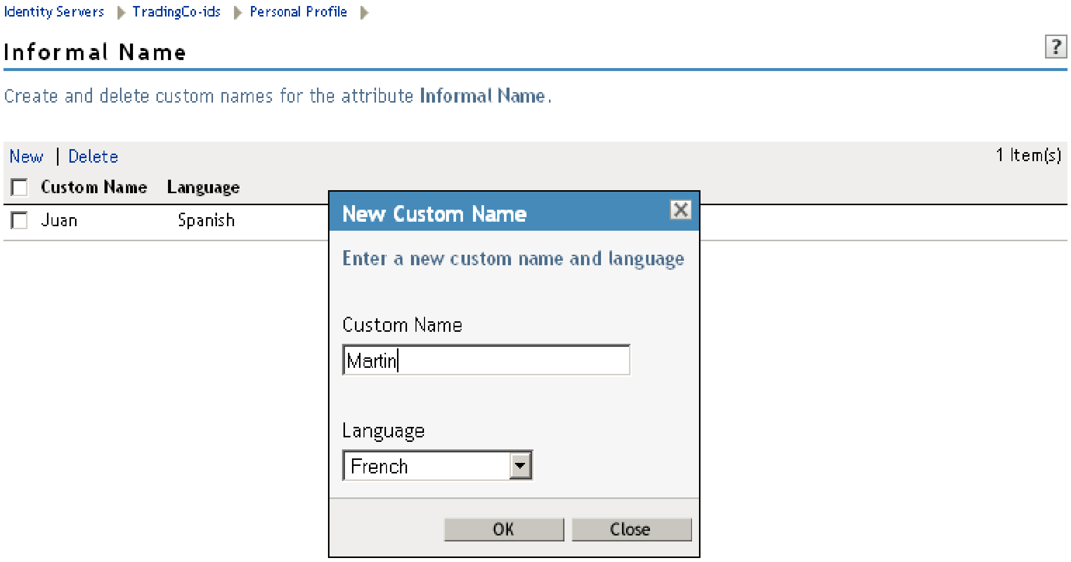
8 Update the Identity Server configuration on the Servers page.

12.6 Customizing Attribute Names

You can change the display name of an attribute names for the Credential, Custom, Employee, and Personal profiles. The customized names are displayed on the My Profile page in the User Portal. The users see the custom names applicable to their language. Custom Attributes are displayed on the My Profile page in the User Portal in place of the corresponding English attribute name when the language in the drop-down list is the accepted language of the browser.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Provider > [Profile] > Custom Attribute Names*.

2 Click the data item name to view the customized attribute names.


Identity Servers ▶ TradingCo-ids ▶ Personal Profile ▶

Informal Name

Create and delete custom names for the attribute **Informal Name**.

New | Delete — 1 Item(s)

☐	Custom Name	Language
☐	Juan	Spanish

New Custom Name

Enter a new custom name and language

Custom Name: Martin

Language: French

3 Click *New* to create a new custom name.

4 Type the name and select a language.

5 Click *OK*.

6 On the Custom Attribute Names page, click *OK*.

7 On the Web Service Provider page, click *OK*.

8 Update the Identity Server configuration on the Servers page.

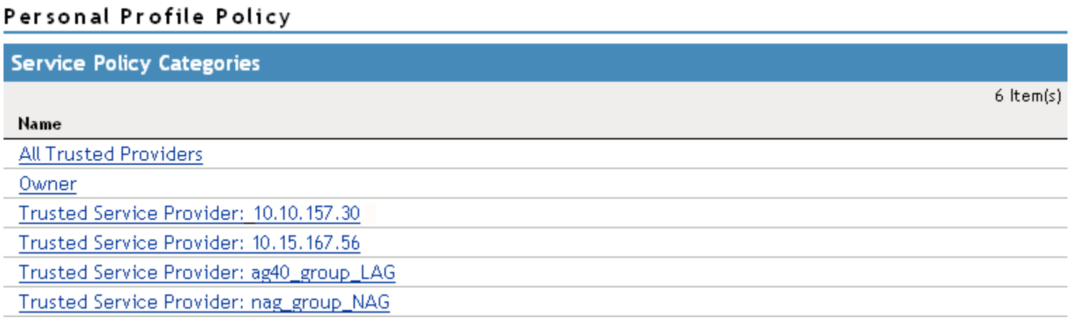
12.7 Editing Web Service Policies

Web Service Policies are permission policies (query and modify) that govern how identity providers share end-user data with service providers. Administrators and policy owners (users) can control whether private information is always allowed to be given, never allowed, or must be requested.

As an administrator, you can configure this information for the policy owner, for specific service providers, or globally for all service providers. You can also specify what policies are displayed for the end user in the User Portal, and whether users are allowed to edit them.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Provider.*

2 Click the *Policy* link next to the service name.


Personal Profile Policy

Service Policy Categories — 6 Item(s)

Name
All Trusted Providers
Owner
Trusted Service Provider: 10.10.157.30
Trusted Service Provider: 10.15.167.56
Trusted Service Provider: ag40_group_LAG
Trusted Service Provider: nag_group_NAG

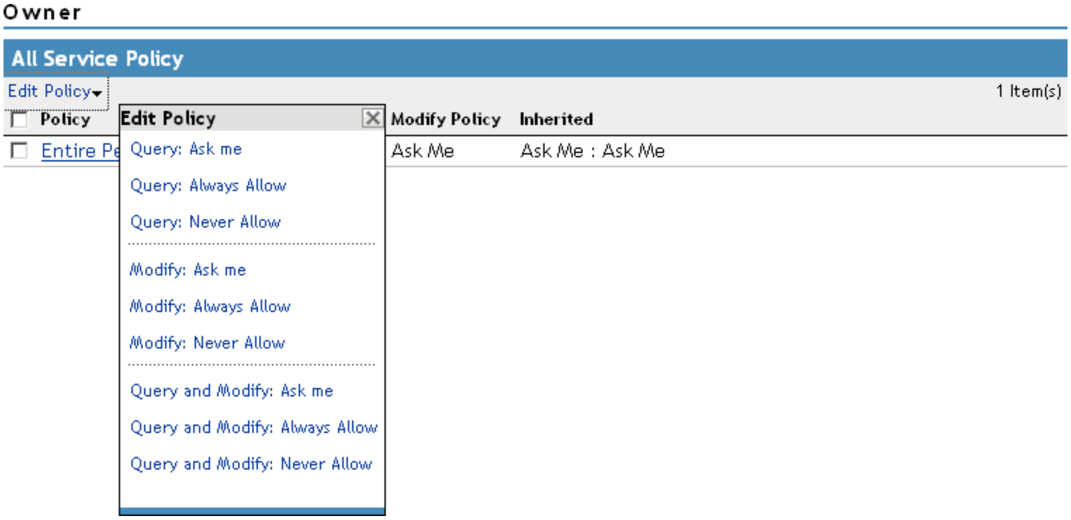
3. Click the category you want to edit.

   All Trusted Providers: Policies that are defined by the service provider's ability to query and modify the particular Liberty attributes or groups of attributes for the Web service. When All Trusted Providers permissions are established, and a service provider needs data, the system first looks here to determine whether user data is allowed, never allowed, or must be asked for. If no solution is found in All Trusted Providers, the system examines the permissions established within the specific service provider.

   Owners: Policies that limit the end user's ability to modify or query data from his or her own profile. The settings you specify in the *Owner* group are reflected on the My Profile page in the User Portal. Portal users have the authority to modify the data items in their profiles. The data items include Liberty and LDAP attributes for personal identity, employment, and any customized attributes defined in the Identity Server configuration. Any settings you specify in the Administration Console override what is displayed in the User Portal. Overrides are displayed in the *Inherited* column.

   If you want the user to have Write permission for a given data item, and that data item is used in an LDAP Attribute Map, then you must configure the LDAP Attribute Map with Write permission.

4. On the All Service Policy page, select the policy's check box, then click *Edit Policy*.



   This lets you modify the parent service policy attribute. Any selections you specify on this page are inherited by child policies.

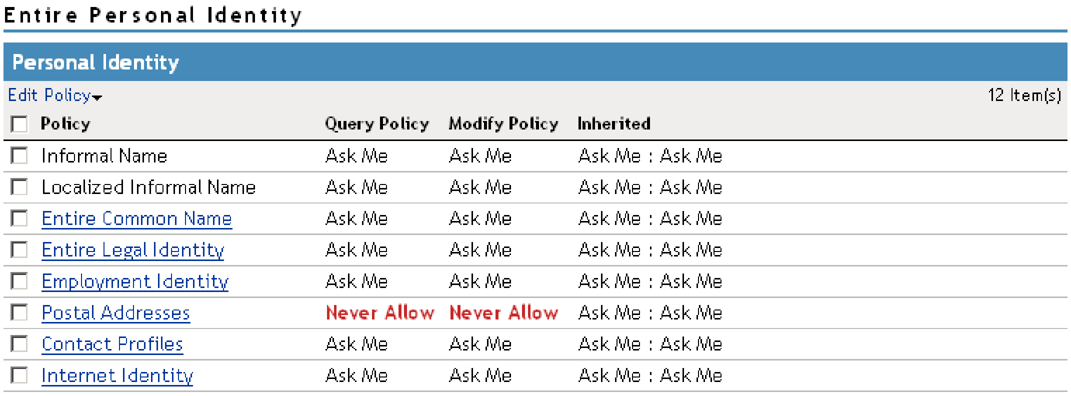
   Query Policy: Allows the service provider to query for the data on a particular attribute. This is similar to read access to a particular piece of data.

   Modify Policy: Allows the service provider to modify a particular attribute. This is similar to write access to a particular piece of data.

   Query and Modify: Allows you to set both options at once.

5. To edit child attributes of the parent, click the policy.

   In the following example, child attributes are inheriting Ask Me permission from the parent *Entire Personal Identity* attribute. The *Postal Address* attribute, however, is modified to never allow permission for sharing.



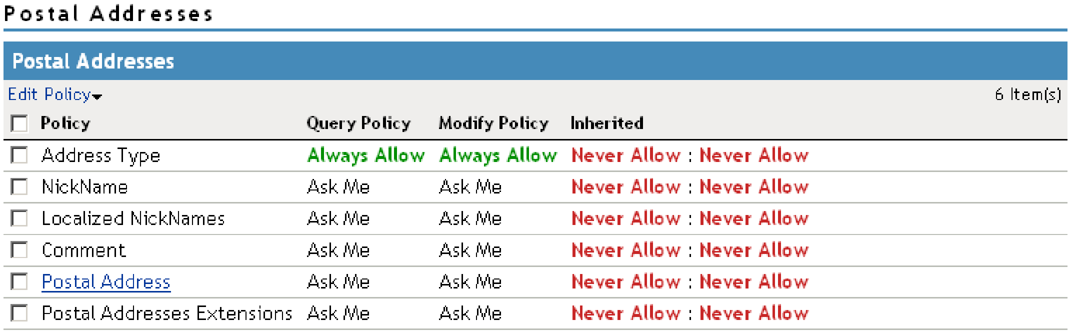
Entire Personal Identity

Personal Identity

Edit Policy ▾ 12 Item(s)

☐ Policy	Query Policy	Modify Policy	Inherited
☐ Informal Name	Ask Me	Ask Me	Ask Me : Ask Me
☐ Localized Informal Name	Ask Me	Ask Me	Ask Me : Ask Me
☐ Entire Common Name	Ask Me	Ask Me	Ask Me : Ask Me
☐ Entire Legal Identity	Ask Me	Ask Me	Ask Me : Ask Me
☐ Employment Identity	Ask Me	Ask Me	Ask Me : Ask Me
☐ Postal Addresses	Never Allow	Never Allow	Ask Me : Ask Me
☐ Contact Profiles	Ask Me	Ask Me	Ask Me : Ask Me
☐ Internet Identity	Ask Me	Ask Me	Ask Me : Ask Me

If you click the *Postal Address* attribute, all of its child attributes have inherited the *Never Allow* setting. You can specify different permission attributes for *Address Type* (for example), but the inherited policy still overrides changes made at the child level, as shown below.



Postal Addresses

Postal Addresses

Edit Policy ▾ 6 Item(s)

☐ Policy	Query Policy	Modify Policy	Inherited
☐ Address Type	Always Allow	Always Allow	Never Allow : Never Allow
☐ NickName	Ask Me	Ask Me	Never Allow : Never Allow
☐ Localized NickNames	Ask Me	Ask Me	Never Allow : Never Allow
☐ Comment	Ask Me	Ask Me	Never Allow : Never Allow
☐ Postal Address	Ask Me	Ask Me	Never Allow : Never Allow
☐ Postal Addresses Extensions	Ask Me	Ask Me	Never Allow : Never Allow

The interface allows these changes in order to simplify switching between configurations if, for example, you want to remove an inherited policy.

Inherited: Specifies the settings inherited from the parent attribute policy, when you view a child attribute. In the User Portal, settings displayed under *Inherited* are not modifiable by the user. At the top-level policy in the User Portal, the values are inherited from the settings in the Administration Console. Thereafter, inheritance can come from the service policy or the parent data item's policy.

Ask Me: Specifies that the service provider requests from the user what action to take.

Always Allow: Specifies that the identity provider always allows the attribute data to be sent to the service provider.

Never Allow: Specifies that the identity provider never allows the attribute data to be sent to the service provider.

When a request for data is received, the Identity Server examines policies to determine what action to take. For example, if a service provider like DigitalAirlines.com requires a postal address for the user, the Identity Server performs the following actions:

- Checks the settings specified in *All Service Providers*.
- If no solution is found, checks for the policy settings configured for the service provider.

6 Click *OK* until the Web Service Provider page is displayed.

7 Click *OK*, then update the Identity Server as prompted.

12.8 Configuring the Web Service Consumer

The Web service consumer is the component within the identity provider that request attributes from Web service providers. The identity provider and Web services consumer cooperate to redirect the user or resource owner to the identity provider, allowing interaction. You can configure an interaction service, which allows the identity provider to pose simple questions to a user. This service can be offered by trusted Web services consumers, or by a dedicated interaction service provider that has a reliable means of communication with the users.

1. In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Liberty > Web Service Consumers*

   The following general settings configure time limits and processing speed:

   Protocol Timeout (seconds): Limits the time the transport protocol allows.

   Provider Timeout (seconds): Limits the request processing at the Web service provider. This value must always be equal to or longer than the *Provider Timeout* value.

   Attribute Cache Enabled: A subsystem of the Web service consumer that caches attribute data that the Web service consumer requests. For example, if the Web service consumer has already requested a first name attribute from a Web service provider, the Web service consumer does not need to request the attribute again. This setting improves performance when enabled. However, you can disable this option to increase system memory.

2. Specify how and when the identity provider interacts with the user:

   Always Allow Interaction: Allows interaction to take place between users and service providers.

   Never Allow Interaction: Never allows interaction between users and service providers.

   Always Allow Interaction for Permissions, Never for Data: Allows interaction for permissions, never for data.

   Maximum Allowed Interaction Time: Specifies the allowed time (in seconds).

3. To specify the allowable methods that a Web service provider can use for user interaction, click one of the following options:

   Redirect to a User Interaction Service: Allows the Web service consumer to redirect the user agent to the Web service provider to ask questions. After the Web service provider has obtained the information it needs, it can redirect the user back to the Web service consumer.

   Call a Trusted User Interaction Service: Allows the Web service provider to trust the Web service consumer to act as proxy for the resource owner.

4. Under *Security Settings*, fill in the following fields:

   WSS Security Token Type: Instructs the Web service consumer/requestor how to place the token in the security header as outlined in the Liberty ID-WSF Security Mechanisms.

   Signature Algorithm: The signature algorithm to use for signing the payload.

5. Click *OK*, then update the Identity Server configuration as prompted.

12.9 Mapping LDAP and Liberty Attributes

You can create an LDAP attribute map or edit an existing one. Attribute mapping involves specifying how single-value and multi-value data items map to single-value and multi-value LDAP attributes. A single-value attribute can contain no more than one value, and a multi-value attribute can contain more than one. An example of a single-value attribute might be a person's gender, and an example of a multi-value attribute might be a person's various e-mail addresses, phone numbers, or titles.

The following fields are common among all attribute maps and are defined here:

Type: Specifies the map type. Access Manager comes with a predefined "one-to-one" mapping type for the Liberty profiles of Personal, Employee, and General. However, the following sections describe how to create additional map types:

- Section 12.9.1, "Configuring One-to-One Attribute Maps," on page 185
- Section 12.9.2, "Configuring Employee Type Attribute Maps," on page 188
- Section 12.9.3, "Configuring Employee Status Attribute Maps," on page 189
- Section 12.9.4, "Configuring Postal Address Attribute Maps," on page 190
- Section 12.9.5, "Configuring Contact Method Attribute Maps," on page 192
- Section 12.9.6, "Configuring Gender Attribute Maps," on page 193
- Section 12.9.7, "Configuring Marital Status Attribute Maps," on page 194

Name: The name you want to give the map.

Description: A description of the map.

Access Rights: A drop-down menu that provide the broadest control for the page. If you set this to *Read/Write*, you can specify rights for individual data items.

In order for user provisioning to succeed, you must select *Read/Write* from the *Access Rights* drop-down menu for any maps that use an attribute during user provisioning.

User Stores: The user store that a map applies to. If a user logs into a user store that is not in the map's user store list, that map is not used to read or write attributes for that user.

LDAP Attribute Name: The LDAP attribute name that you want to map to the Liberty attribute.

LDAP Attribute Value: The predefined LDAP attribute values that you want to map to the Liberty values. These LDAP values are those you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value. Values must match the attribute as it appears in the directory exactly. For example, "givenName" must be entered as "givenName" in the text field or the mapping does not work.

12.9.1 Configuring One-to-One Attribute Maps

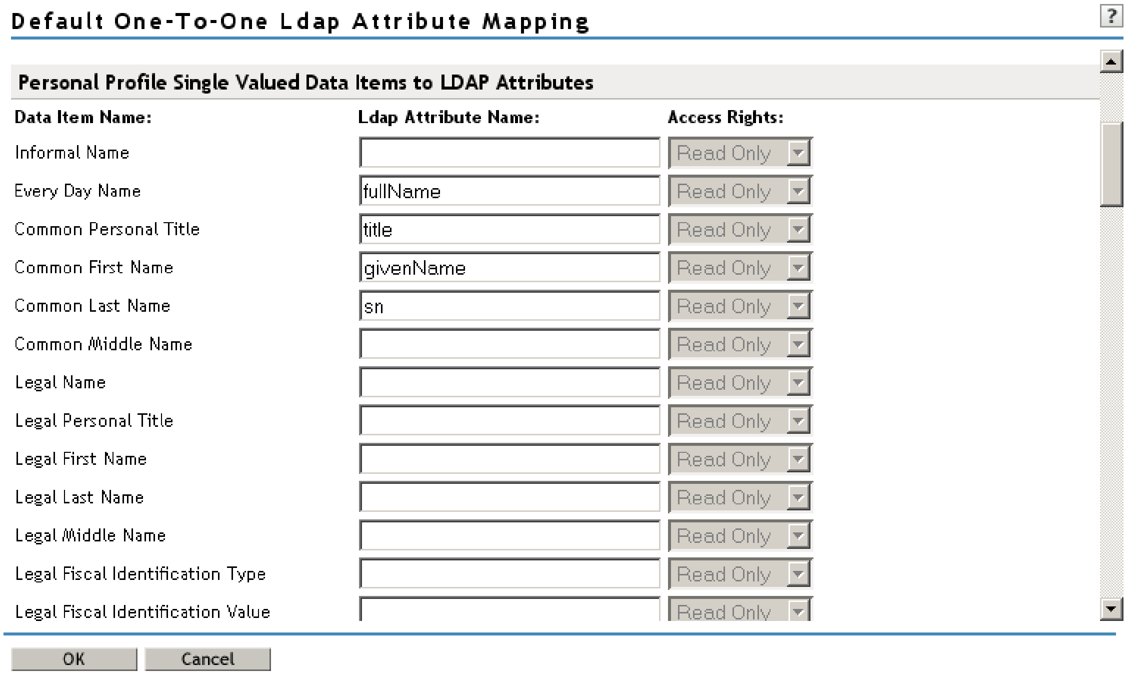
A one-to-one map enables you to map single-value and multiple-value LDAP attribute names to standard Liberty attributes. A default one-to-one attribute map is provided with Access Manager, but you can also define your own.

An example of a one-to-one attribute map might be the single-valued Liberty attribute Common Name (CommonName) used by the Personal Profile that is mapped to the LDAP attribute givenName. The attribute value CN might be mapped to the LDAP fullName. You can further configure the various Liberty values to map to any LDAP attribute names that you use.

1. In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Liberty > LDAP Attribute Mapping > New > One to One*.
2. Configure the map using the following guidelines:
   - Mapping Personal Profile Single-Value Data Items to LDAP Attributes
   - Mapping Personal Profile Multiple-Value Data Items to LDAP Attributes
   - Mapping Employee Profile Single-Value Data Items to LDAP Attributes
   - Mapping Employee Profile Multiple-Value Data Items to LDAP Attributes
   - Mapping Custom Profile Single-Value Data Items to LDAP Attributes
   - Mapping Custom Profile Multiple-Value Data Items to LDAP Attributes
3. After you create the mapping, click *Finish*.
4. On the LDAP Attribute Mapping page, click *OK*.
5. Update the Identity Server configuration on the Servers page as prompted.

Mapping Personal Profile Single-Value Data Items to LDAP Attributes

The data items displayed are single-value Liberty Personal Profile attributes that you can map to the single-valued LDAP attributes that you have defined for your directory.



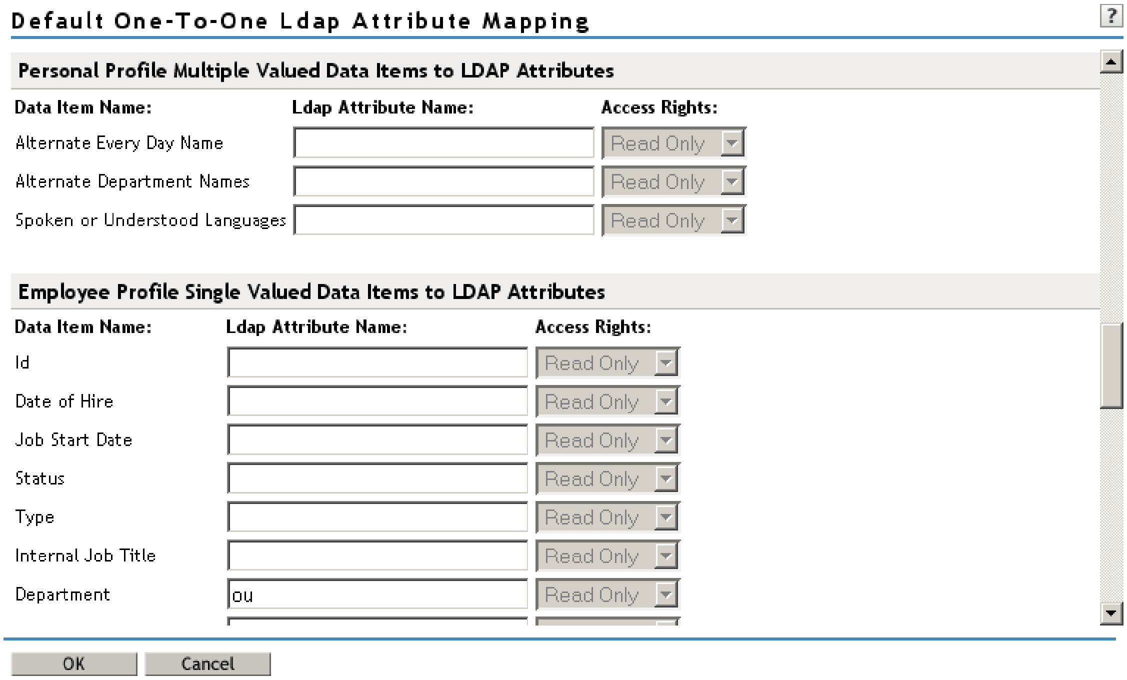
Default One-To-One Ldap Attribute Mapping

Personal Profile Single Valued Data Items to LDAP Attributes

Data Item Name:	Ldap Attribute Name:	Access Rights:
Informal Name		Read Only
Every Day Name	fullName	Read Only
Common Personal Title	title	Read Only
Common First Name	givenName	Read Only
Common Last Name	sn	Read Only
Common Middle Name		Read Only
Legal Name		Read Only
Legal Personal Title		Read Only
Legal First Name		Read Only
Legal Last Name		Read Only
Legal Middle Name		Read Only
Legal Fiscal Identification Type		Read Only
Legal Fiscal Identification Value		Read Only

Mapping Personal Profile Multiple-Value Data Items to LDAP Attributes

Use the fields on this page to map multiple-value attributes from the Liberty Personal Profile to the multiple-value LDAP attributes you have defined for your directory. For example, you can map the Liberty attribute Alternate Every Day Name (AltCN) to the LDAP attribute you have defined for this purpose in your directory.


Default One-To-One Ldap Attribute Mapping

Personal Profile Multiple Valued Data Items to LDAP Attributes

Data Item Name:	Ldap Attribute Name:	Access Rights:
Alternate Every Day Name		Read Only
Alternate Department Names		Read Only
Spoken or Understood Languages		Read Only

Employee Profile Single Valued Data Items to LDAP Attributes

Data Item Name:	Ldap Attribute Name:	Access Rights:
Id		Read Only
Date of Hire		Read Only
Job Start Date		Read Only
Status		Read Only
Type		Read Only
Internal Job Title		Read Only
Department	ou	Read Only

Mapping Employee Profile Single-Value Data Items to LDAP Attributes

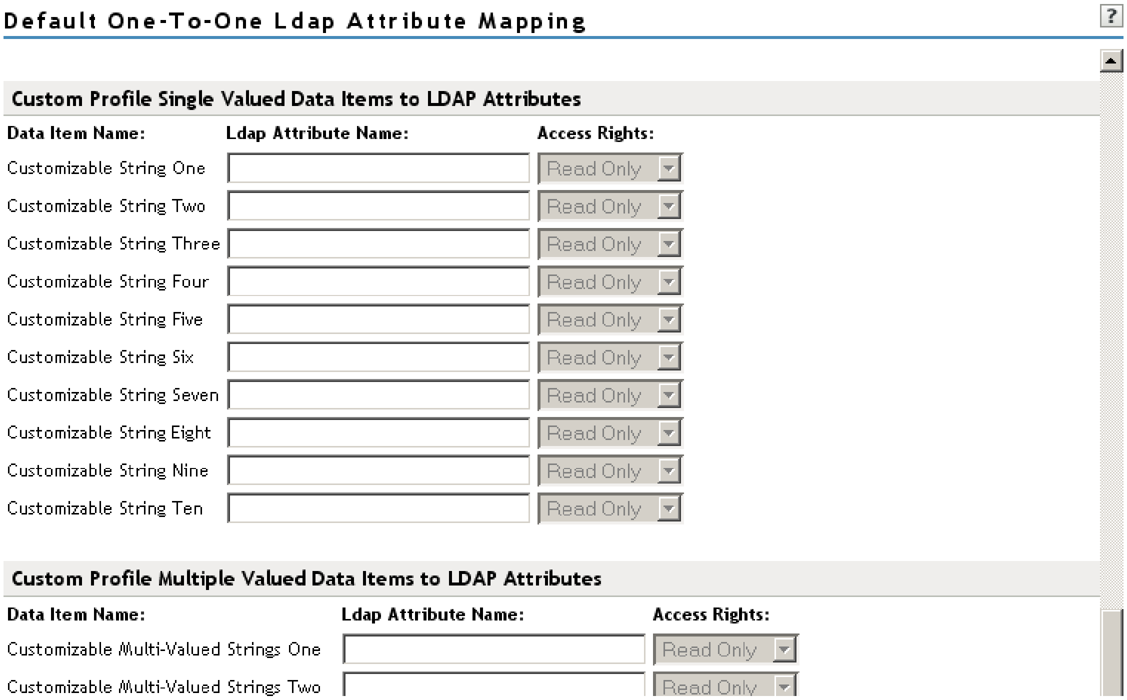
Map the Liberty Employee Profile single-value attributes to the LDAP attributes you have defined in your directory for entries such as ID, Date of Hire, Job Start Date, Department, and so on.

Mapping Employee Profile Multiple-Value Data Items to LDAP Attributes

Map the Liberty Employee Profile multiple-value attributes to the LDAP attributes you have defined in your directory.

Mapping Custom Profile Single-Value Data Items to LDAP Attributes

Map custom Liberty profile single-value attributes to LDAP attributes you have defined in your directory. These attributes are customizable strings associated with the Custom Profile.



Default One-To-One Ldap Attribute Mapping

Custom Profile Single Valued Data Items to LDAP Attributes

Data Item Name:	Ldap Attribute Name:	Access Rights:
Customizable String One		Read Only
Customizable String Two		Read Only
Customizable String Three		Read Only
Customizable String Four		Read Only
Customizable String Five		Read Only
Customizable String Six		Read Only
Customizable String Seven		Read Only
Customizable String Eight		Read Only
Customizable String Nine		Read Only
Customizable String Ten		Read Only

Custom Profile Multiple Valued Data Items to LDAP Attributes

Data Item Name:	Ldap Attribute Name:	Access Rights:
Customizable Multi-Valued Strings One		Read Only
Customizable Multi-Valued Strings Two		Read Only

Customizable String (1 - 10): The Custom Profile allows custom single-value and multiple-value attributes to be defined without using the Data Model Extension XML to extend a service's schema. To use a customizable attribute, navigate to the *Custom Attribute Names* tab on the Custom Profile Details page (see Section 12.6, "Customizing Attribute Names," on page 180). There you can customize the name of any of the predefined single-value or multiple-value customizable attributes in the Custom Profile. After you customize a name, you can use that attribute in the same way you use any other profile attribute.

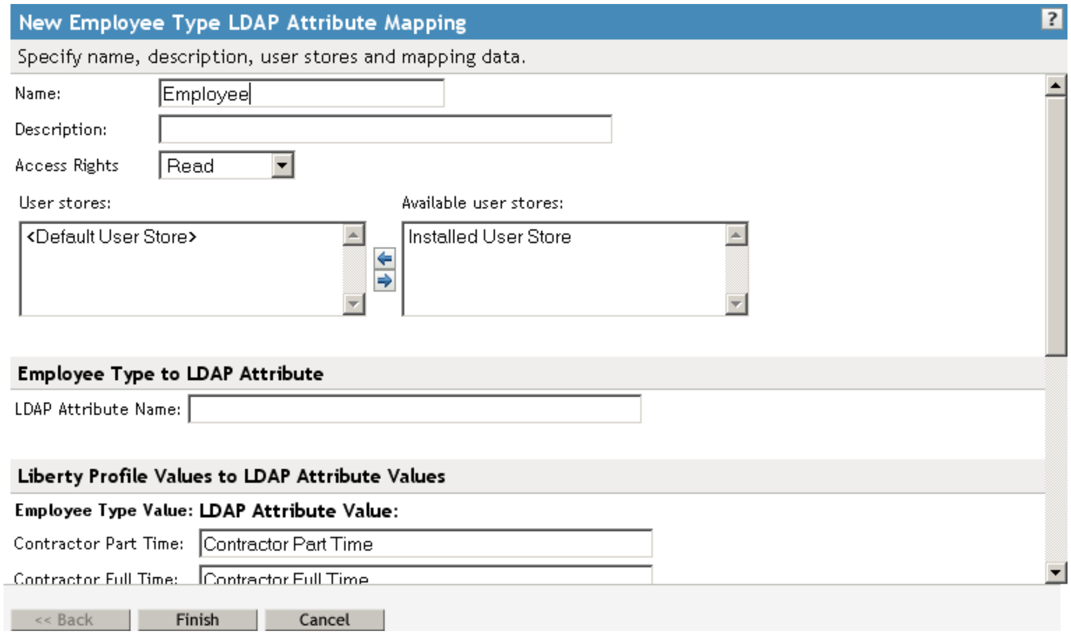
Mapping Custom Profile Multiple-Value Data Items to LDAP Attributes

Customizable Multi-Valued Strings (1 - 5): Similar to customizable strings for single-value attributes, except these attributes can have multiple values. Use this list of fields to map directory attributes that can have multiple values (like SN) to multiple-value strings from the Custom Profile.

12.9.2 Configuring Employee Type Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Employee Type. This is an Employee Profile attribute. Examples of Liberty values appended to this attribute include Contractor Part Time, Contractor Full Time, Full Time Regular, and so on.

1. In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Liberty* > *LDAP Attribute Mapping* > *New* > *Employee Type*.



New Employee Type LDAP Attribute Mapping

Specify name, description, user stores and mapping data.

Name: Employee

Description:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Employee Type to LDAP Attribute

LDAP Attribute Name:

Liberty Profile Values to LDAP Attribute Values

Employee Type Value:	**LDAP Attribute Value:**
Contractor Part Time:	Contractor Part Time
Contractor Full Time:	Contractor Full Time

2 Specify a name and description for the map.

3 Choose the type of access rights you want.

Select *Read/Write* for any attributes used in user provisioning.

4 In the *LDAP Attribute Name* field, type the LDAP attribute name that you want to map to the Liberty Employee Type attribute.

5 In the *LDAP Attribute Value* fields, type your predefined LDAP attribute values that you want to map to the *Liberty Employee Type* values.

These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.

6 Click *Finish*.

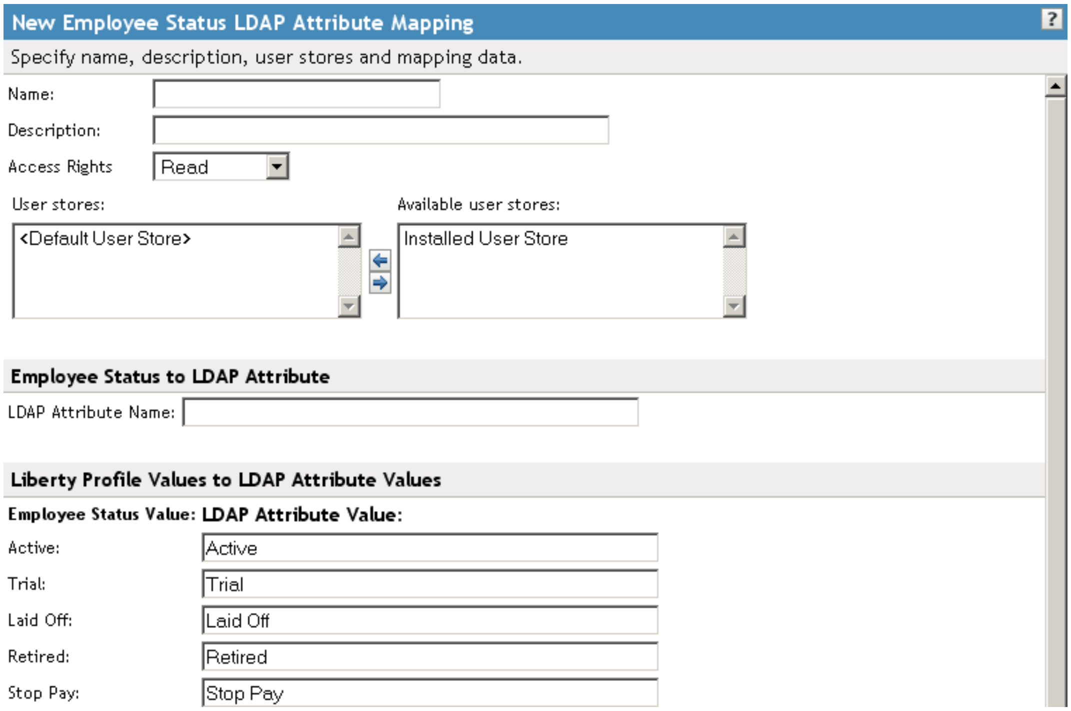
7 On the LDAP Attribute Mapping page, click *OK*.

8 Update the Identity Server configuration on the Servers page as prompted.

12.9.3 Configuring Employee Status Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Employee Status. This is an Employee Profile attribute. Examples of the values appended to this Liberty attribute include Active, Trial, Retired, Terminated, and so on.

1 In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Liberty > LDAP Attribute Mapping > New > Employee Status*.



New Employee Status LDAP Attribute Mapping

Specify name, description, user stores and mapping data.

Name:

Description:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Employee Status to LDAP Attribute

LDAP Attribute Name:

Liberty Profile Values to LDAP Attribute Values

Employee Status Value:	LDAP Attribute Value:
Active:	Active
Trial:	Trial
Laid Off:	Laid Off
Retired:	Retired
Stop Pay:	Stop Pay

2 Specify a name and description for the map.

3 Choose the type of access rights you want.

Select *Read/Write* for any attributes used in user provisioning.

4 In the *LDAP Attribute Name* field, type the LDAP attribute name that you want to map to the *Liberty Employee Status* element.

5 In the *LDAP Attribute Value* fields, type the predefined LDAP attribute values that you want to map to the *Liberty Employee Status* values.

These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.

6 Click *Finish*.

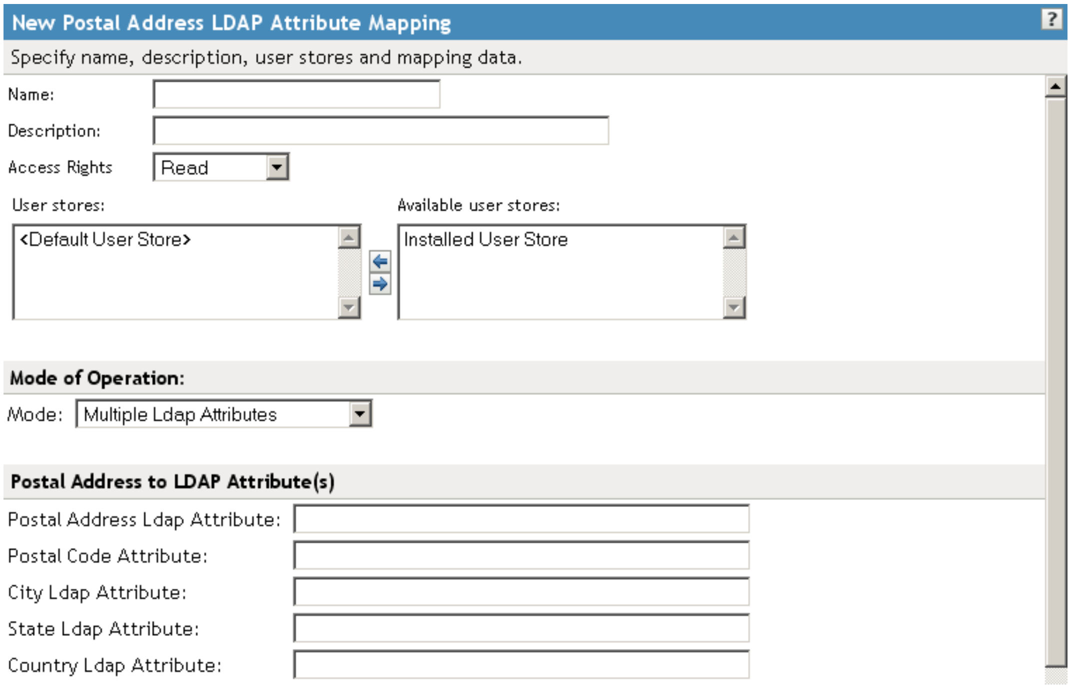
7 On the LDAP Attribute Mapping page, click *OK*.

8 Update the Identity Server configuration on the Servers page as prompted.

12.9.4 Configuring Postal Address Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for Postal Address. The PostalAddress element refers to the local address, including street or block with a house number, and so on. This is a Personal Profile attribute.

1 In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Liberty > LDAP Attribute Mapping > New > Postal Address*.


New Postal Address LDAP Attribute Mapping
Specify name, description, user stores and mapping data.
Name:
Description:
Mode of Operation:
Postal Address to LDAP Attribute(s)
Postal Address Ldap Attribute:
Postal Code Attribute:
City Ldap Attribute:
State Ldap Attribute:
Country Ldap Attribute:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Mode: Multiple Ldap Attributes

2 Specify a name and description for the map.

3 Choose the type of access rights you want.

Select *Read/Write* for any attributes used in user provisioning.

4 In the *Mode* drop-down menu, select either *Multiple LDAP Attributes* or *Single Deliminated LDAP Attributes*.

Multiple LDAP Attributes: Allows you to map multiple LDAP attributes to multiple Liberty Postal Address elements. When you select this option, the following Liberty Postal Address elements are displayed under the *Postal Address to LDAP Attributes* group. Type the LDAP attributes that you want to map to the Liberty elements.

- Postal Address
- Postal Code
- City
- State
- Country

Single Deliminated LDAP Attributes: Allows you to specify one LDAP attribute that is used to hold multiple elements of a Liberty Postal Address in a single delimited value. When you select this option, the page displays the following fields:

- **Delimited LDAP Attribute Name:** The delimited LDAP attribute name you have defined for the LDAP postal address that you want to map to the Liberty Postal Address attribute.
- **Delimiter:** The character to use to delimit single-value entries. A $ sign is the default delimiter.

5 (Multiple LDAP Attributes mode) Under *Postal Address Template Data*, fill in the following options:

Nickname: (Required) A Liberty element name used to identify the Postal Address object.

Contact Method Type: Select the contact method type, such as *Domicile*, *Work*, *Emergency*, and so on.

6 (Single Deliminated LDAP Attributes mode) Under *One-Based Field Position in Delimited LDAP Attribute*, specify the order in which the information is contained in the string. Select 1 for the value that comes first in the string, 2 for the value that follows the first delimiter, etc.

7 Click *Finish*.

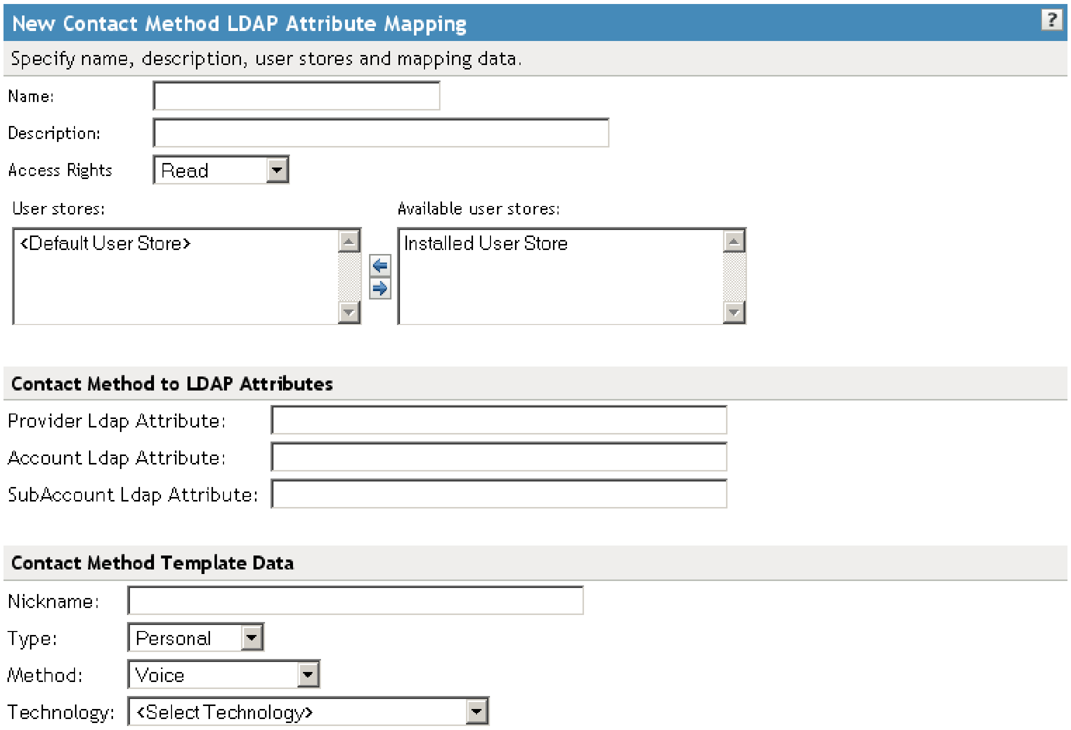
8 On the LDAP Attribute Mapping page, click *OK*.

9 Update the Identity Server configuration on the Servers page as prompted.

12.9.5 Configuring Contact Method Attribute Maps

You can map the LDAP attribute you have defined for contact methods to the Liberty attribute Contact Method (MsgContact).

1 In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Liberty > LDAP Attribute Mapping > New > Contact Method*.



New Contact Method LDAP Attribute Mapping

Specify name, description, user stores and mapping data.

Name:

Description:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Contact Method to LDAP Attributes

Provider Ldap Attribute:

Account Ldap Attribute:

SubAccount Ldap Attribute:

Contact Method Template Data

Nickname:

Type: Personal

Method: Voice

Technology: <Select Technology>

2 Specify a name and description for the map.

3 Choose the type of access rights you want.

Select *Read/Write* for any attributes used in user provisioning.

4 Under *Contact Method to LDAP Attributes*, fill in the following fields to map to the Liberty Contact Method attribute:

Provider LDAP Attribute: Maps to the Liberty attribute MsgProvider, which is the service provider or domain that provides the messaging service.

Account LDAP Attribute: Maps to the Liberty attribute MsgAccount, which is the account or address information within the messaging provider.

SubAccount LDAP Attribute: Maps to the Liberty MsgSubaccount, which is the subaccount within a messaging account, such as voice mail box associated with a phone number.

5 Under *Contact Method Template Data*, specify the settings for the Liberty attribute values of:

Nickname: Maps to the Liberty attribute Nick, which is an informal name for the contact.

Type: Maps to the Liberty attribute MsgType (such as Mobile, Personal, or Work).

Method: Maps to the Liberty MsgMethod (such as Voice, Fax, or E-mail).

Technology: Maps to the Liberty attribute MsgTechnology (such as Pager, VOIP, and so on).

6 Click *Finish*.

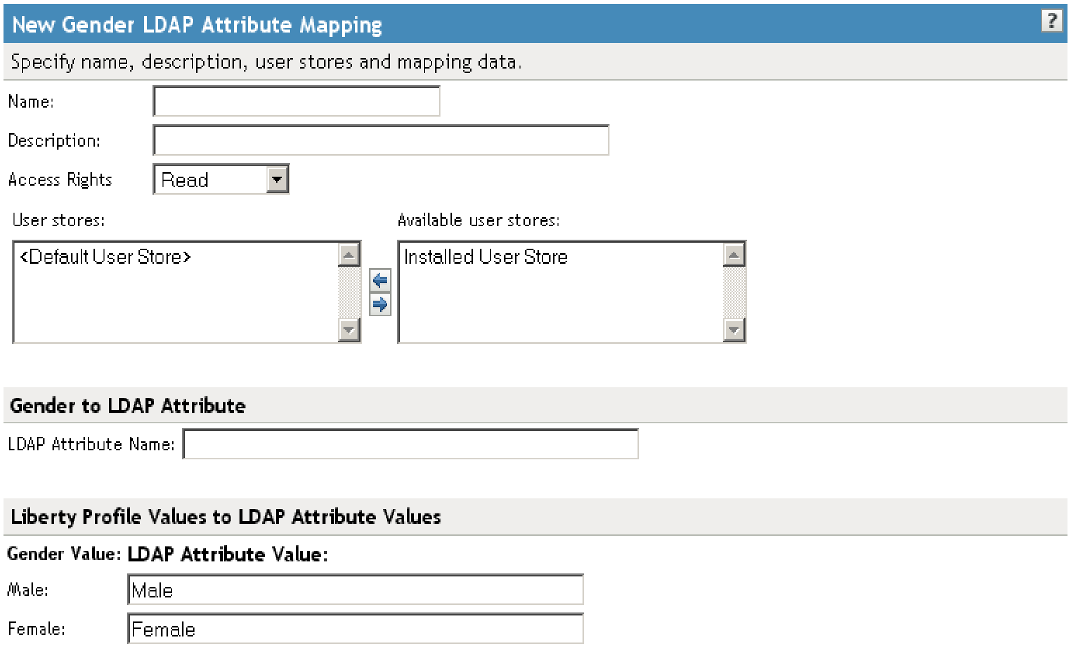
7 On the LDAP Attribute Mapping page, click *OK*.

8 Update the Identity Server configuration on the Servers page as prompted.

12.9.6 Configuring Gender Attribute Maps

You can map the LDAP attribute name and values to the Liberty profile values for the Gender attribute. You can use gender to differentiate between people with the same name, especially in countries where national ID numbers cannot be collected. This is a Personal Profile attribute.

1 In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Liberty* > *LDAP Attribute Mapping* > *New* > *Gender*.



New Gender LDAP Attribute Mapping

Specify name, description, user stores and mapping data.

Name:

Description:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Gender to LDAP Attribute

LDAP Attribute Name:

Liberty Profile Values to LDAP Attribute Values

Gender Value:	LDAP Attribute Value:
Male:	Male
Female:	Female

2 Specify a name and description for the map.

3 Choose the type of access rights you want.

Select *Read/Write* for any attributes used in user provisioning.

4 In the *LDAP Attribute Name* field, type the LDAP attribute name that you want to map to the Liberty element Gender.

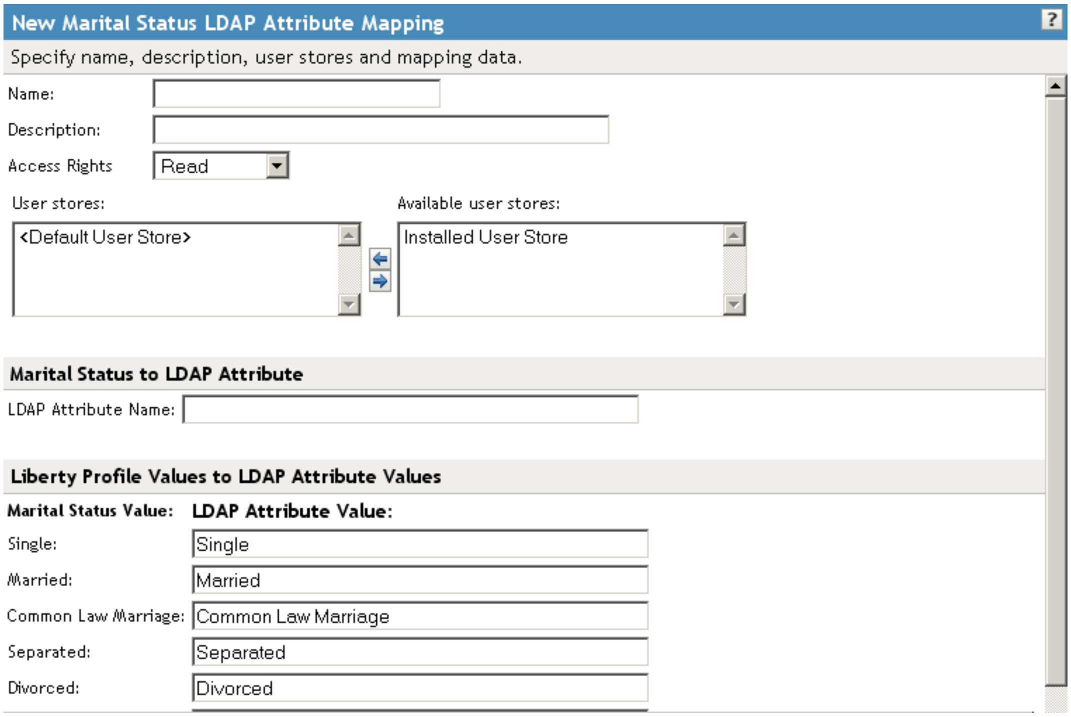
5. In the *LDAP Attribute Value* fields, type your predefined LDAP attribute values that you want to map to the Gender values.

   These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.
6. Click *Finish*.
7. On the LDAP Attribute Mapping page, click *OK*.
8. Update the Identity Server configuration on the Servers page as prompted.

12.9.7 Configuring Marital Status Attribute Maps

You can map the LDAP marital status attribute to the Liberty attribute. The Liberty Marital Status (MaritalStatus) element includes appended values such as single, married, divorced, and so on. For example, `urn:liberty:id-sis-pp:maritalstatus:single`. This is a Personal Profile attribute.

1. In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Liberty > LDAP Attribute Mapping > New > Marital Status*.



New Marital Status LDAP Attribute Mapping

Specify name, description, user stores and mapping data.

Name:

Description:

Access Rights: Read

User stores: <Default User Store>

Available user stores: Installed User Store

Marital Status to LDAP Attribute

LDAP Attribute Name:

Liberty Profile Values to LDAP Attribute Values

Marital Status Value:	LDAP Attribute Value:
Single:	Single
Married:	Married
Common Law Marriage:	Common Law Marriage
Separated:	Separated
Divorced:	Divorced

2. Specify a name and description for the map.
3. Choose the type of access rights you want.

   Select *Read/Write* for any attributes used in user provisioning.
4. In the *LDAP Attribute Name* field, type the LDAP attribute name that you want to map to the Liberty element Marital Status (MaritalStatus).
5. In the *LDAP Attribute Value* fields, type your predefined LDAP attribute values that you want to map to the MaritalStatus values.

These are the values that you want to store in the LDAP attribute for each given Liberty attribute value. The LDAP attribute map then maps the actual Liberty URI value, back and forth, to this supplied value.

6. Click *Finish*.
7. On the LDAP Attribute Mapping page, click *OK*.
8. Update the Identity Server configuration on the Servers page as prompted.

Access Gateway Configuration

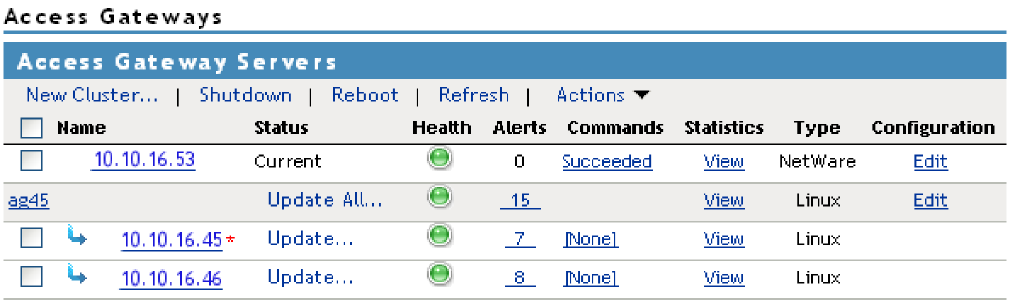
III

This section describes how you configure and manage the Novell® Access Gateway. The procedures in this section assume that you have already done the following:

- Installed the Access Gateway. (See *Novell Access Manager 3.0 SP3 IR2 Installation Guide*).
- Logged in to the Administration Console as the admin user. (See "Logging In to the Administration Console" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.)
- Created an Identity Server configuration. (See Chapter 6, "Configuring an Identity Server," on page 55.)

You should be familiar with the steps documented in "Setting Up a Basic Access Manager Configuration" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*. These guides are designed to work together.

In the Administration Console when you click *Access Manager* > *Access Gateways*, the following page appears.



Access Gateways

Access Gateway Servers

☐	Name	Status	Health	Alerts	Commands	Statistics	Type	Configuration
☐	10.10.16.53	Current	●	0	Succeeded	View	NetWare	Edit
ag45		Update All...	●	15		View	Linux	Edit
☐	↳ 10.10.16.45*	Update...	●	7	[None]	View	Linux	
☐	↳ 10.10.16.46	Update...	●	8	[None]	View	Linux	

The links on this page allow you to manage the Access Gateways on your network. The following sections describe these tasks.

- Chapter 13, "Configuring the Access Gateway to Protect Web Resources," on page 199
- Chapter 14, "Configuring the Access Gateway for SSL," on page 239
- Chapter 15, "Server Configuration Settings," on page 251
- Chapter 16, "Configuring the Cache Settings," on page 279
- Chapter 17, "Protecting Multiple Resources," on page 293

For monitoring tasks such as auditing, logging, statistics, health, command status, and alerts, see Part VII, "Monitoring Access Manager Components," on page 503.

For information on creating a fault tolerant system, including clustering Access Gateways, see "Clustering and Fault Tolerance" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

For security planning, see Chapter 1, "Security Considerations," on page 25.

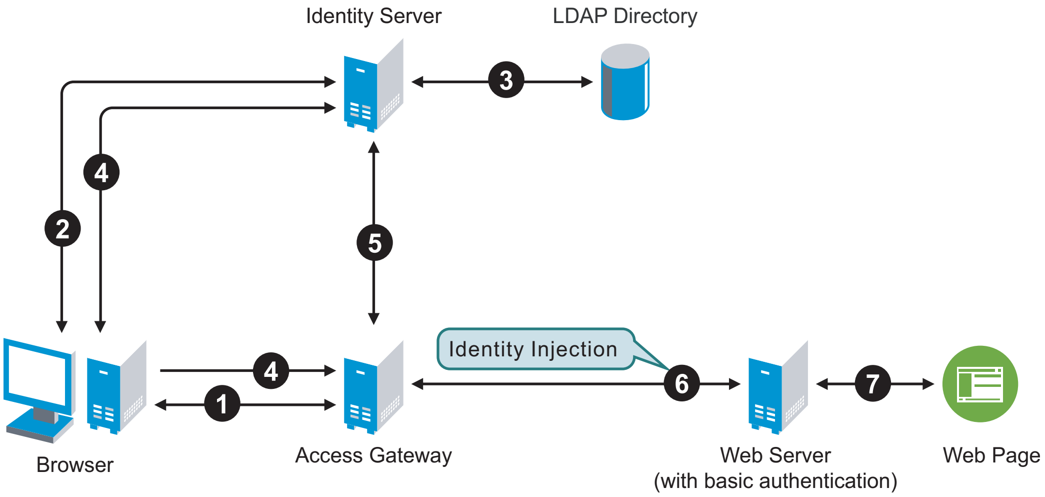
Configuring the Access Gateway to Protect Web Resources

13

The Novell® Access Gateway is a reverse proxy server (protected site server) that restricts access to Web-based content, portals, and Web applications that employ authentication and access control policies. It also provides single sign-on to multiple Web servers and Web applications by securely providing the credential information of authenticated users to the protected servers and applications. The Access Gateway lets you simplify, secure, and accelerate your Internet business initiatives.

A typical Access Manager configuration includes an Identity Server with LDAP directories and an Access Gateway with a protected Web server. Figure 13-1 illustrates the process flow that allows an authorized user to access the protected resource on the Web server.

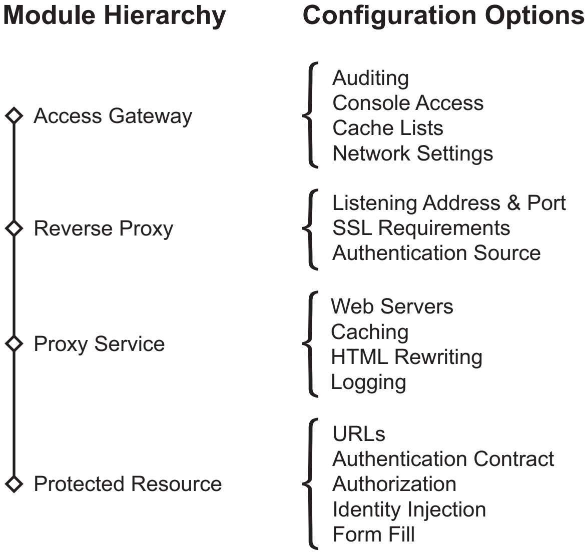
Figure 13-1 *Accessing a Web Resource*



1. The user requests access to a resource protected by the Access Gateway.
2. The Access Gateway redirects the user to the Identity Server, which prompts the user for a username and password.
3. The Identity Server verifies the username and password against an LDAP directory (eDirectory™, Active Directory, or Sun ONE).
4. The Identity Server returns an authentication success to the browser and the browser forwards the resource request to the Access Gateway.
5. The Access Gateway verifies that the user is authenticated and retrieves the user’s credentials from the Identity Server.
6. The Access Gateway uses an Identity Injection policy to insert the basic authentication credentials in the HTTP header of the request and sends it to the Web server.
7. The Web server grants access and sends the requested page to the user.

When you are setting up the Access Gateway to protect Web resources, you create and configure reverse proxies, proxy services, and protected resources. The following figure illustrates the hierarchy of these modules and the major configuration tasks you perform on each module.

Figure 13-2 *Access Gateway Modules and Their Configuration Options*



This hierarchy allows you to have precise control over what is required to access a particular resource, while at the same time allowing you to provide a single sign-on solution for all the resources protected by the Access Gateway. The authentication contract and the Authorization, Identity Injection, and Form Fill policies are configured at the resource level so that you can enable exactly what the resource requires. This allows you to decide where access decisions are made:

- You can configure the Access Gateway to control access to the resource.
- You can configure the Web server for access control and configure the Access Gateway to supply the required information.
- You can use the first method for some resources and the second method for other resources or use both methods on the same resource.

This section describes the following tasks:

- Section 13.1, "Creating a Reverse Proxy and Proxy Service," on page 200
- Section 13.2, "Configuring a Proxy Service," on page 204
- Section 13.3, "Configuring the Web Servers of a Proxy Service," on page 205
- Section 13.4, "Configuring Protected Resources," on page 207
- Section 13.5, "Configuring HTML Rewriting," on page 215
- Section 13.6, "Configuring Connection and Session Limits," on page 233

13.1 Creating a Reverse Proxy and Proxy Service

A reverse proxy acts as the front end to your Web servers on your Internet or intranet and off-loads frequent requests, thereby freeing up bandwidth. The proxy also increases security because the IP addresses of your Web servers are hidden from the Internet.

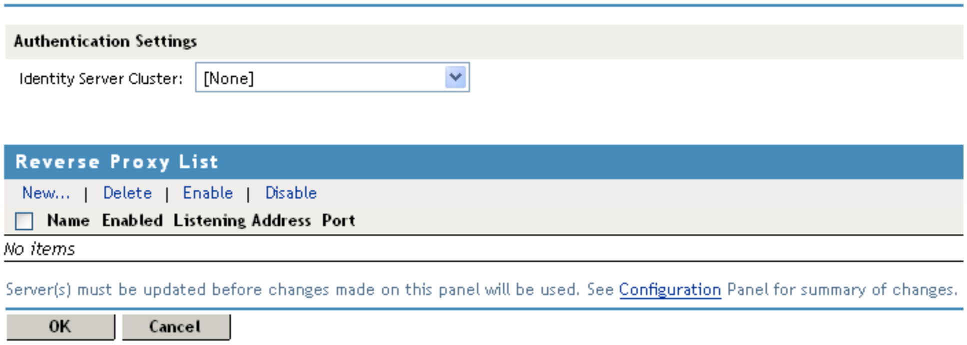
To create a reverse proxy, you must create at least one proxy service with a protected resource. You must supply a name for each of these components. Reverse proxy names and proxy service names must be unique to the Access Gateway because they are configured for global services such as IP addresses and TCP ports. For example, if you have a reverse proxy named `products` and another reverse proxy named `library`, only one of these reverse proxies can have a proxy service named `corporate`.

Protected resource names need to be unique to the proxy service, but they don't need to be unique to the Access Gateway because they are always accessed through their proxy service. For example, if you have a proxy service named `account` and a proxy service named `sales`, they both can have a protected resource named `public`.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit*

The *Edit* link is either for a single Access Gateway or for a cluster of Access Gateways.

2 Click *Reverse Proxy / Authentication*.



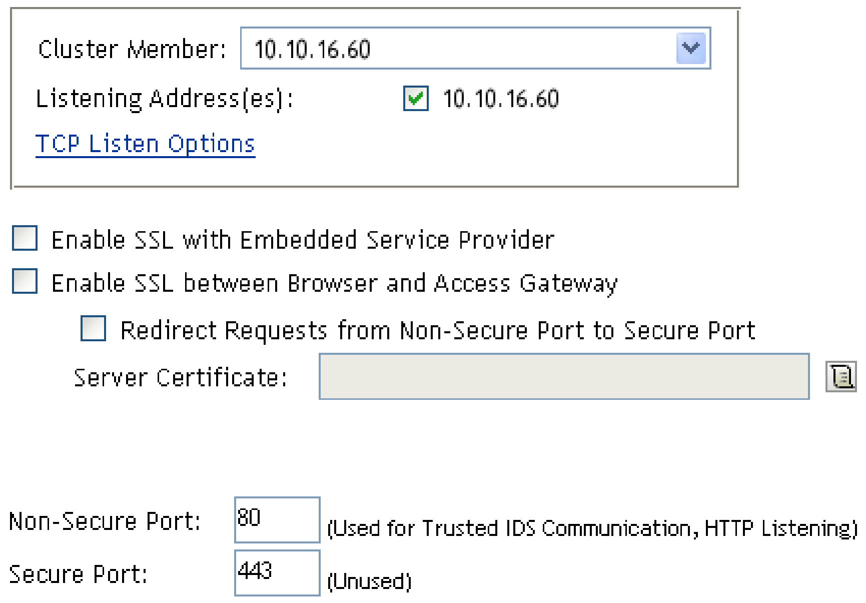
Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

3 Select an *Identity Server Cluster*.

Identity Server Cluster: Specifies the Identity Server you want the Access Gateway to trust for authentication. Select the configuration you have assigned to the Identity Server.

Whenever an Identity Server is assigned to a new trust relationship, the Identity Server needs to be updated. This process is explained following the step that saves this configuration setting (see Step 5 on page 205 and Step 6 on page 205).

4 In the *Reverse Proxy List*, click *New*, specify a display name for the reverse proxy, then click *OK*.



Cluster Member: 10.10.16.60

Listening Address(es): ☑ 10.10.16.60

TCP Listen Options

☐ Enable SSL with Embedded Service Provider

☐ Enable SSL between Browser and Access Gateway

☐ Redirect Requests from Non-Secure Port to Secure Port

Server Certificate:

Non-Secure Port: 80 (Used for Trusted IDS Communication, HTTP Listening)

Secure Port: 443 (Unused)

5 Enable a listening address. Fill in the following fields:

Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. The *Listening Address(es)* and *TCP Listen Options* modifications apply to the selected server. Modifications made to any other options on the page apply to all servers in the cluster.

Listening Address(es): Displays a list of available IP addresses. If the server has only one IP address, only one is displayed and it is automatically selected. If the server has multiple addresses, you can select one or more IP addresses to enable. You must enable at least one address by selecting its check box.

If the Access Gateway is in a cluster, you must select a listening address for each cluster member.

TCP Listen Options: Provides options for configuring how requests are handled between the reverse proxy and the client browsers. You cannot set up the listening options until you create and configure a proxy service. For information about these options, see Section 13.6.1, "Configuring TCP Listen Options for Clients," on page 234.

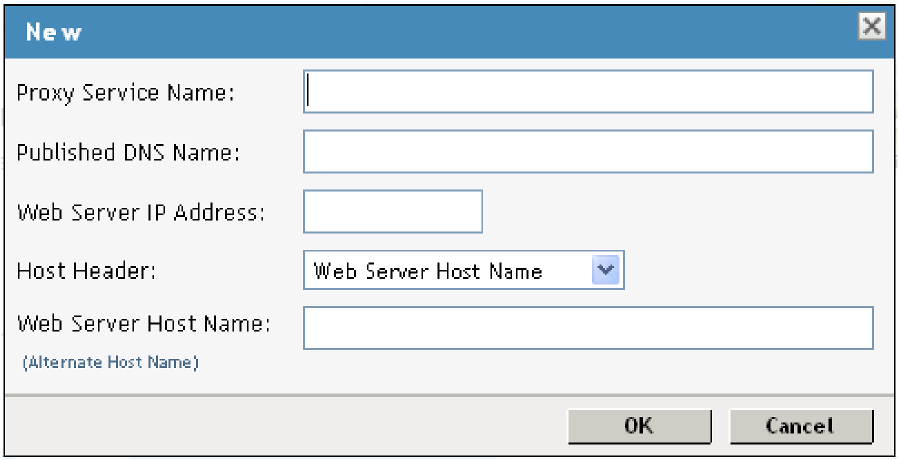
6 Configure the listening ports:

Non-Secure Port: Specifies the port on which to listen for HTTP requests; the default port for HTTP is 80. Depending upon your configuration, this port might also handle other tasks. These tasks are listed to the right of the text box.

Secure Port: Specifies the port on which to listen for HTTPS requests; the default port for HTTPS is 443.

For information about the SSL options, see Chapter 14, "Configuring the Access Gateway for SSL," on page 239.

7 In the *Proxy Service List* section, click *New*.



The first proxy service of a reverse proxy is considered the master (or parent) proxy. Subsequent proxy services can use domain-based, path-based, or virtual multi-homing, relative to the published DNS name of the master proxy service. If you are creating a second proxy service for a reverse proxy, see Section 17.2, "Using Multi-Homing to Access Multiple Resources," on page 295.

8 Fill in the fields:

Proxy Service Name: Specify a display name for the proxy service, which the Administration Console uses for its interfaces.

Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address.

Web Server IP Address: Specify the IP address of the Web server you want this proxy service to manage. You can specify additional Web server IP addresses by clicking the Web Server Addresses link when you have finished creating the proxy service.

Host Header: Specify whether the HTTP header should contain the name of the back-end Web server (*Web Server Host Name* option) or whether the HTTP header should contain the published DNS name (the *Forward Received Host Name* option).

Web Server Host Name: Specify the DNS name of the Web server that the Access Gateway should forward to the Web server. If you have set up a DNS name for the Web server and it requires its DNS name in the HTTP header, specify that name in this field. If the Web server has absolute links referencing its DNS name, include this name in this field. If you selected *Forward Received Host Name*, this option is not available.

NOTE: For iChain® administrators, the *Web Server Host Name* is the alternate host name when configuring a Web Server Accelerator.

---

9 Click *OK*.

10 Continue with Section 13.2, "Configuring a Proxy Service," on page 204 or select one of the following tasks:

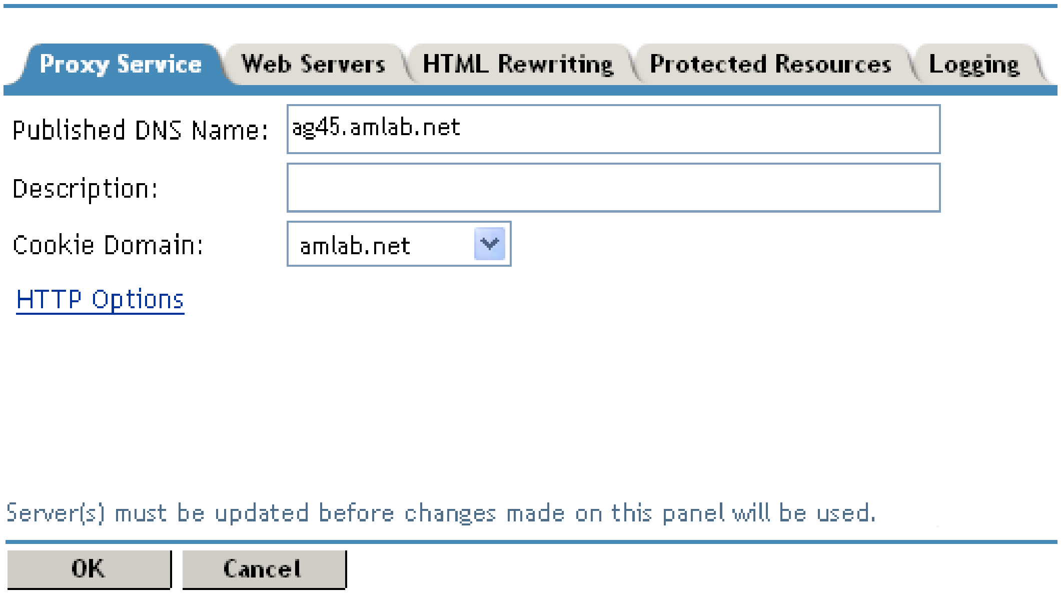
- For instructions on creating multiple reverse proxies, see Section 17.3, "Managing Multiple Reverse Proxies," on page 304.
- For instructions on creating multiple proxy services for a reverse proxy, see Section 17.2, "Using Multi-Homing to Access Multiple Resources," on page 295.

13.2 Configuring a Proxy Service

A reverse proxy can have multiple proxy services, and each proxy service can protect multiple resources. You can modify the following features of the proxy service:

- Web servers
- HTML rewriting
- Logging
- Protected resources
- Caching

1 To configure a proxy service, click *Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service]*.



Published DNS Name: ag45.amlab.net

Description:

Cookie Domain: amlab.net

Server(s) must be updated before changes made on this panel will be used.

2 Fill in the following fields:

Published DNS Name: Displays the value that users are currently using to access this proxy service. This DNS name must resolve to the IP address you set up as a listening address on the Access Gateway. You should modify this field only if you have modified the DNS name you want users to use to access this resource.

This name determines the possible values of the *Cookie Domain*.

Description: (Optional). Provides a field where you can describe the purpose of this proxy service or specify any other pertinent information.

Cookie Domain: Specifies the domain for which the cookie is valid.

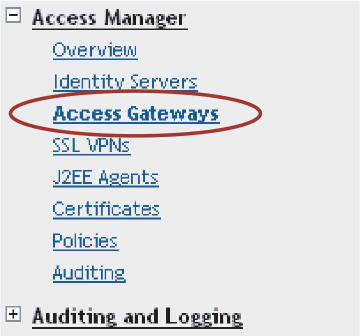
If one proxy service has a DNS name of www.support.novell.com and the second proxy service has a DNS name of www.developernet.novell.com, the cookie domains are support.novell.com for the first proxy service and developernet.novell.com for the second proxy service. You can configure them to share the same cookie domain by selecting novell.com for each proxy service. Single sign-on between the proxy services is simplified when they proxy services share the same cookie.

HTTP Options: Allows you to set up global caching and custom caching options for this proxy service. See the following:

- Section 16.2, "Controlling Browser Caching," on page 282
- Section 16.3, "Configuring Custom Cache Control Headers," on page 283
- Section 16.1, "Configuring Global Caching Options," on page 279

3 Click *OK* to save your changes to browser cache.

4 Click the *Access Gateways* link.



5 To apply your changes, click *Update* > *OK*.

Until this step, nothing has been permanently saved or applied. The *Update* status pushes the configuration to the server and writes the configuration to the configuration data store. When the update has completed successfully, the server returns the status of *Current*.

To save the changes to the configuration store without applying them, do not click *Update*. Instead, click *Edit*. On the Configuration page, click *OK*. The *OK* button on this pages saves the cached changes to the configuration store. The changes are not applied until you click *Update* on the Access Gateways page.

6 Update the Identity Server to accept the new trusted relationship. Click *Identity Servers* > *Update*.

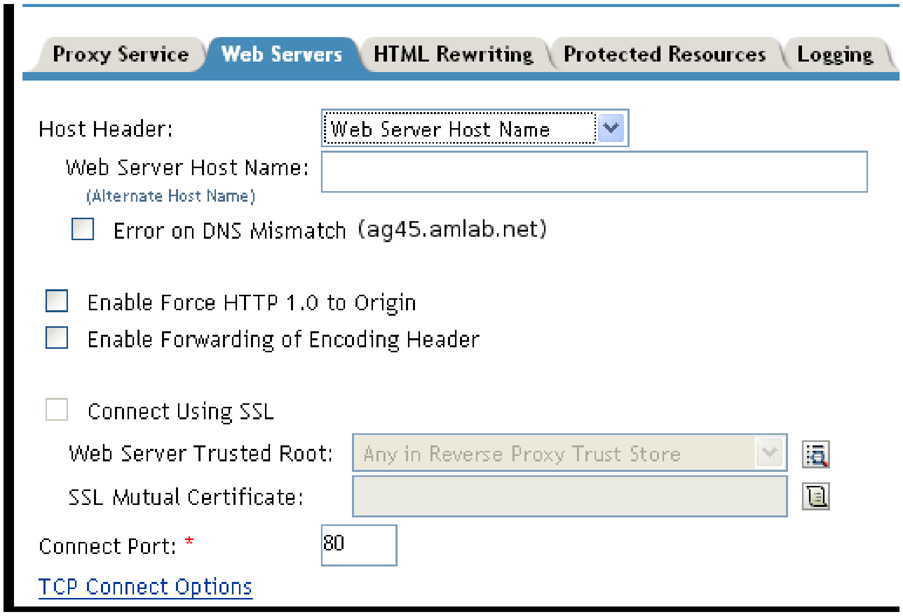
7 Continue with one of the following.

- If the Web server that contains the resources you want to protect does not use the standard HTML port (port 80), you need to configure the Web server. See Section 13.3, "Configuring the Web Servers of a Proxy Service," on page 205.
- Until you configure a protected resource, the proxy service blocks access to all services on the Web server. To configure a protected resource, see Section 13.4, "Configuring Protected Resources," on page 207.

13.3 Configuring the Web Servers of a Proxy Service

The Web server configuration determines how the Access Gateway handles connections and packets between itself and the Web servers.

1 Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Web Servers*.



Host Header: Web Server Host Name

Web Server Host Name:

(Alternate Host Name)

☐ Error on DNS Mismatch (ag45.amlab.net)

☐ Enable Force HTTP 1.0 to Origin

☐ Enable Forwarding of Encoding Header

☐ Connect Using SSL

Web Server Trusted Root: Any in Reverse Proxy Trust Store

SSL Mutual Certificate:

Connect Port: * 80

TCP Connect Options

2 Specify the host name that is placed in the HTTP header of the packets being sent to the Web servers. In the *Host Header* field, select one of the following:

- **Forward Received Host Name:** Indicates that you want the HTTP header to contain the published DNS name that the user sent in the request.
- **Web Server Host Name:** Indicates that you want the published DNS name that the user sent in the request to be replaced by the DNS of the Web server. Use the *Web Server Host Name* field to specify this name.

3 Select *Error on DNS Mismatch* to have the proxy determine whether the proxy service should compare the host name in the DNS header that came from the browser with the DNS name specified in the *Web Server Host Name* option.The value in the parentheses is the value that comes in the header from the browser.

If you enable this option and the names don't match, the request is not forwarded to the Web server. Instead, the proxy service returns an error to the requesting browser. This option is only available when you select to send the *Web Server Host Name* in the HTTP header.

4 If your browsers are capable of sending HTTP 1.1 requests, configure the following fields to match your Web servers.

Enable Force HTTP 1.0 to Origin: Indicates whether HTTP 1.1 requests from browsers are translated to HTTP 1.0 requests before sending them to the Web server. If your browsers are sending HTTP 1.1 requests and your Web server can only handle HTTP 1.0 requests, you should enable this option.

When the option is enabled, the Access Gateway translates an HTTP 1.1 request to an HTTP 1.0 request.

Enable Forwarding of Encoding Header: Determines whether the HTTP 1.1 header is sent to the Web server:

- If you enable this option, the entire HTTP 1.1 header is sent to the Web server. If your browsers are sending HTTP 1.1 requests and your Web servers are HTTP 1.1 compliant, this is the configuration you should use.

- If you enable this option and you have also enabled the *Enable Force HTTP 1.0 to Origin* option, a few select fields of the HTTP 1.1 header (such as the content encoding header for compression) are sent to the Web server. If your Web server is not HTTP 1.1-compliant, but it can handle a few HTTP 1.1 fields, you should enable this option.
- If your Web server can handle only HTTP 1.0 headers, you should not enable this option.

5 To enable SSL connections between the proxy service and its Web servers, select *Connect Using SSL*. For configuration information for this option, *Web Server Trusted Root*, and *SSL Mutual Certificate*, see Section 14.4, "Configuring SSL between the Proxy Service and the Web Servers," on page 244.

6 In the *Connect Port* field, specify the port that the Access Gateway should use to communicate with the Web servers. The following table lists some default port values for common types of Web servers.

Server Type	**Non-Secure Port**	**Secure Port**
Web server with HTML content	80	443
SSL VPN	8080	8443
WebSphere*	9080	9443
JBoss*	8080	8443

7 To control how idle and unresponsive Web server connections are handled and to optimize these processes for your network, select *TCP Connect Options*. For more information, see Section 13.6.2, "Configuring TCP Connect Options for Web Servers," on page 235.

8 To add a Web server, click *New* in the *Web Server List* and specify the IP address or the fully qualifier DNS name of the Web server.

The Web servers added to this list must contain identical Web content. Configuring your system with multiple servers with the same content adds fault tolerance and increases the speed for processing requests. For more information about this process, see Section 17.1, "Setting Up a Group of Web Servers," on page 294.

9 To delete a Web server, select the Web server, then click *Delete*.

This deletes the Web server from the list so that the Access Gateway no longer sends requests to the deleted Web server. At least one Web server must remain in the list. You must delete the proxy service to remove the last server in the list.

10 To save your changes to browser cache, click *OK*.

11 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.4 Configuring Protected Resources

A protected resource configuration specifies the directory (or directories) on the Web server that you want to protect. The protected resource configuration specifies the authorization contract and the policies that should be used to enforce protection. The authentication contract and the policies (Authorization, Identity Injection, and Form Fill) enable the single sign-on environment for the user. The type of protections a resource requires depends upon the resource, the Web server, and the conditions you define for the resource.

You can select from the following types of protection:

Authentication Contract: Specifies the type of credentials the user must use to log in (such as name and password or secure name and password). You can select *None* for the contract, which allows the resource to be a public resource, with no login required.

Authorization Policy: Specifies the conditions a user must meet to be allowed access to a protected resource. You define the conditions, and the Access Gateway enforces the Authorization policies. For example, you can assign roles to your users, and use these roles to grant and deny access to resources.

Identity Injection Policy: Specifies the information that must be injected into the HTTP header. If the Web application has been configured to look for certain fields in the header and the information cannot be found, the Web application determines whether the user is denied access or redirected. The Web application defines the requirements for Identity Injection. The Identity Injection policies allow you to inject the required information into the header.

Form Fill Policy: Allows you to manage forms that Web servers return in response to client requests. Form fill allows you to prepopulate fields in a form on first login and then securely save the information in the completed form to a secret store for subsequent logins. The user is prompted to reenter the information only when something changes, such as a password.

These policies allow you to design a custom policy for each protected resource:

- Resources that share the same protection requirements can be configured as a group. You set up the policies, and then add the URLs of each resource that requires these policies.
- A resource that has specialized protection requirements can be set up as a single protected resource. For example, a page that uses Form Fill is usually set up as a single protected resource.

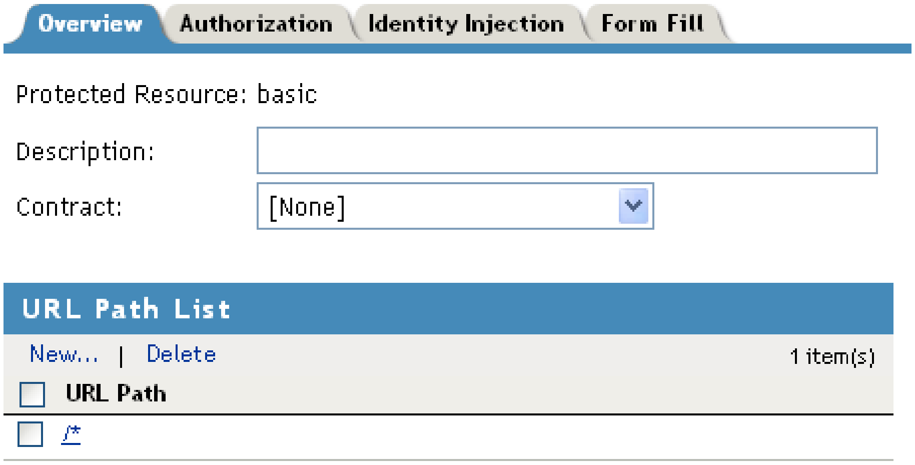
This section describes the following tasks:

- Section 13.4.1, “Setting Up a Protected Resource,” on page 208
- Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210
- Section 13.4.3, “Assigning an Identity Injection Policy to a Protected Resource,” on page 211
- Section 13.4.4, “Assigning a Form Fill Policy to a Protected Resource,” on page 213
- Section 13.4.5, “Assigning a Policy to Multiple Protected Resources,” on page 214

13.4.1 Setting Up a Protected Resource

To configure a protected resource:

1. Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Protected Resources*.
2. Either click the name of an existing resource or click *New*, then specify a display name for the resource.



Protected Resource: basic

Description:

Contract: [None]

URL Path List

☐ URL Path

☐ /*

3 (Optional) Specify a description for the protected resource. You can use it to briefly describe the purpose for protecting this resource.

4 Select the type of contract, which determines the information a user must supply for authentication. By default, the Administration Console allows you to select from the following contracts and options when specifying whether a resource requires an authentication contract:

- **None:** If you want to allow public access to the resource and not require an authentication contract, select *None*.
- **Any Contract:** If the user has authenticated, allows any contract defined for the Identity Server to be valid, or if the user has not authenticated, prompts the user to authenticate, using the default contract assigned to the Identity Server configuration.
- **Name/Password - Basic:** Specifies basic authentication over HTTP, using a standard login pop-up provided by the Web browser.
- **Name/Password - Form:** Specifies a form-based authentication over HTTP, using the Access Manager login form.
- **Secure Name/Password - Basic:** Specifies basic authentication over HTTPS, using a standard login pop-up provided by the Web browser.
- **Secure Name/Password - Form:** Specifies a form-based authentication over HTTPS, using the Access Manager login form.

You can configure other types of contracts. For more information, see Section 8.4, "Configuring Authentication Contracts," on page 111.

If these default contracts are not available, you have not configured a relationship between the Access Gateway and the Identity Server. See Section 13.1, "Creating a Reverse Proxy and Proxy Service," on page 200.

5 Configure the *URL Path*.

The default path is /*, which indicates everything on the Web server. Modify this if you need to restrict access to a specific directory on your Web server. If you have multiple directories on your Web server that require the same authentication contract and access control, add each directory as a URL path.

- **New:** To add a path, click *New*, specify the path, then click *OK*. For example, to allow access to all the pages in the public directory on the Web server, specify the following path:

  `/public/*`

To allow access to all the files in a directory, but not to the subdirectories and their files, specify the following:

```
/?
/public/?
```

The `/?` allows access to the root directory, but not the subdirectories. The `/public/?` allows access to the files in the public directory, but not the subdirectories.

To use this protected resource to protect a single page, specify the path and the filename. For example, to protect the `login.html` page in the `/login` directory, specify the following:

```
/login/login.html
```

This is the type of URL path you want to specify when you create a Form Fill policy for a protected resource. The *URL Path List* normally contains only this one entry. If you have multiple pages that the Form Fill policy applies to, list each one separately in the list. For optimum speed, you want the Access Gateway to be able to quickly identify the page and not search other pages to see if the policy applies to them.

IMPORTANT: URL paths are case insensitive. If your Web server has two paths (`/public/current` and `/public/Current`), a URL path of `/public/current` matches both.

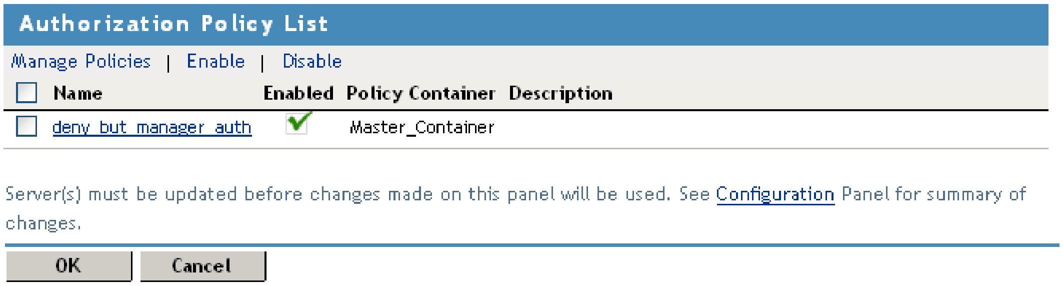
- **Modify:** To modify a path, click the path link, then modify the *URL Path*.
- **Delete:** To delete a path, select the path, then click *Delete*.

6. Click *OK*.
7. In the *Protected Resource List*, ensure that the protected resource you created is enabled.
8. (Optional) To add policies for protecting this resource, continue with one of the following:
   - "Assigning an Authorization Policy to a Protected Resource" on page 210
   - "Assigning an Identity Injection Policy to a Protected Resource" on page 211
   - "Assigning a Form Fill Policy to a Protected Resource" on page 213
   - "Assigning a Policy to Multiple Protected Resources" on page 214
9. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.4.2 Assigning an Authorization Policy to a Protected Resource

An Authorization policy specifies conditions that a user must meet in order to access a resource. The Access Gateway enforces these conditions. The policy can specify the criteria a user must meet either to allow access or to deny access.

1. Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Protected Resources* > *[Name of Protected Resource]* > *Authorization*.



☐ Name	Enabled	Policy Container	Description
☐ deny but manager auth	✓	Master_Container	

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

The *Authorization Policy List* contains all the Access Gateway Authorization policies that have been created on this Administration Console.

2 Select one of the following:

- To enable an existing policy, select the policy, then click *Enable*. Continue with Step 4.
- To disable an existing policy, select the policy, then click *Disable*. Continue with Step 4.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see Section 28.2, "Creating Access Gateway Authorization Policies," on page 420.

  When you have completed your policy modifications, continue with Step 4.
- To create a new policy, click *Manage Policies*. On the Policies page, click *New*, specify a display name, select *Access Gateway: Authorization* as the type, then click *OK*. For configuration information, see Section 28.2, "Creating Access Gateway Authorization Policies," on page 420.

  When you have created your policy, continue with Step 3.

3 To enable the policy you just created, select the policy, then click *Enable*.

Only the policies that are enabled are applied to this resource. All available Authorization policies are listed. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

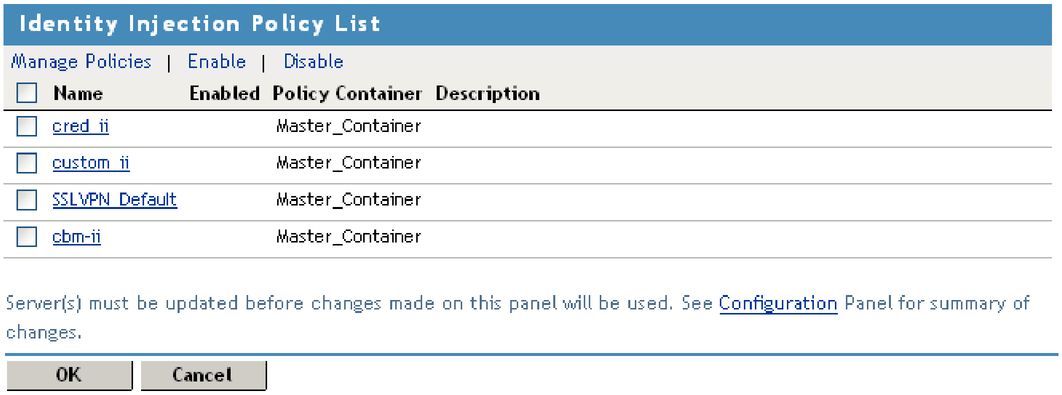
4 To save your changes to browser cache, click *OK*.

5 To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.4.3 Assigning an Identity Injection Policy to a Protected Resource

The Web application defines the requirements for Identity Injection. If a Web application has been configured to look for certain fields in the header and the information cannot be found, the Web application determines whether the user is denied access, granted access, or redirected. You configure an Identity Injection policy to inject into the HTTP header the information that the Web application requires.

1 Click *Access Gateways* > *Edit* > *[Reverse Proxy Name]* > *[Name of Proxy Service]* > *Protected Resources* > *[Name of Protected Resource]* > *Identity Injection*.



Identity Injection Policy List
Manage Policies | Enable | Disable
OK
Cancel

☐ Name	Enabled	Policy Container	Description
☐ cred_ii		Master_Container	
☐ custom_ii		Master_Container	
☐ SSLVPN_Default		Master_Container	
☐ cbm-ii		Master_Container	

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

The *Identity Injection Policy List* contains all the Identity Injection policies that have been created on this Administration Console.

2 Select one of the following:

- To enable an existing policy, select the policy, then click *Enable*. Only the policies that are enabled are applied to this resource. Continue with Step 4.
- To disable an existing policy, select the policy, then click *Disable*. Continue with Step 4.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see Chapter 29, "Creating Identity Injection Policies," on page 467.

  When you have finished your policy modifications, continue with Step 4.
- To create a new policy, click *Manage Policies*. On the Policies page, click *New*, specify a display name, select *Access Gateway: Identity Injection* as the type, then click *OK*. For configuration information, see Chapter 29, "Creating Identity Injection Policies," on page 467.

  When you have created your policy, continue with Step 3.

3 To enable the policy you just created, select the policy, then click *Enable*.

Only the policies that are enabled are applied to this resource. All available Identity Injection policies are listed. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

4 To save your changes to browser cache, click *OK*.

5 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

IMPORTANT: If you enable an Identity Injection policy for a protected resource that has been assigned to use a contract that does not prompt the user for a password and the Identity Injection policy injects the user's password, single sign-on cannot be enabled because the password is not available. To enable single sign-on, you need to use an authentication class that retrieves the user's password and injects it into the user's credentials when the user authenticates using a non-password method such as X509, Radius, smart card, or Kerberos. For information about such a class and how to download and configure it, see Access Management Authentication Class Extension to Retrieve Password for Single Sign-on (http://www.novell.com/communities/node/4556).

13.4.4 Assigning a Form Fill Policy to a Protected Resource

Some client requests cause the Web server to return a form. Sometimes this form contains a request to log in. If you create a Form Fill policy, you can have the Access Gateway fill in the form. When a user first logs in, the Access Gateway prepopulates some fields and prompt the users for the others. The Access Gateway securely saves the information, so that on subsequent logins, the Access Gateway can fill in the form. The user is only prompted to fill in the form when something changes, such as a password expiring.

Form Fill uses two components: the HTML form and the Form Fill policy. The HTML form is created with HTML tags and consists of form elements such as fields, menus, check boxes, and buttons. The Form Fill policy is created by specifying the following:

- Which information is entered automatically and not displayed to the user.
- Which information is displayed so that the user, at least the first time, can enter the information.
- What is done with the information (for example, is it saved so that the user doesn't need to enter it when accessing the form again).

You must create the policy before you can assign it to a resource (see Chapter 30, “Creating Form Fill Policies,” on page 481). To assign a Form Fill policy to a protected resource:

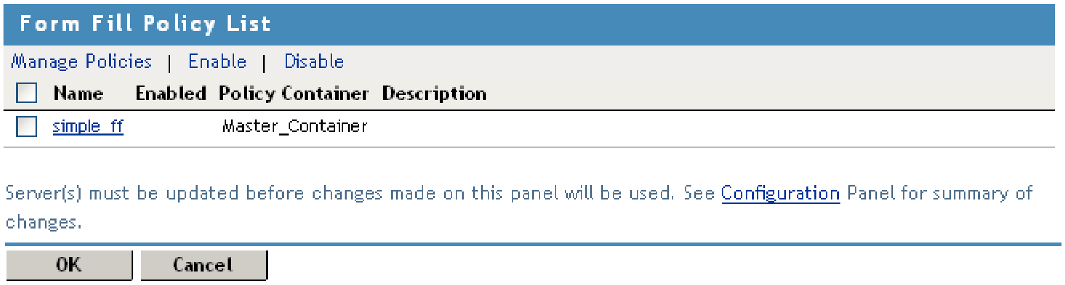
1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Reverse Proxy Name]* > *[Name of Proxy Service]* > *Protected Resources* > *[Name of Protected Resource]*.
2. Examine the entries in the *URL Path List*.

   Ideally, the URL to which you are assigning a Form Fill policy should be a single HTML page or a few HTML pages. If at all possible, it should not be a URL that ends in a wildcard (for example, an asterisk) and therefore matches many pages.

   WARNING: When the URL ends in a wildcard, the Access Gateway must search each page that matches the URL and check to see if it contains the form. This adds extra processing overhead for all the pages that match the URL, but do not contain the form. For more information on the performance problems this can cause, see Section , “Creating a Form Matching Rule,” on page 488.

   ---

3. (Conditional) If the URL is not specific, click the name of the path and modify it.
4. Click *Form Fill*.



☐	Name	Enabled	Policy Container	Description
☐	simple_ff		Master_Container	

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

The *Form Fill Policy List* contains all the Form Fill policies that have been created on this Administration Console.

5 Select one of the following:

- To enable an existing policy, select the policy, then click *Enable*. Only the policies that are enabled are applied to this resource. Continue with Step 7.
- To disable an existing policy, select the policy, then click *Disable*. Continue with Step 7.
- To edit an existing policy, click the name of the policy. Remember that policies can be assigned to multiple protected resources. If you modify the policy, you are also affecting how this policy protects those resources. For configuration information, see Chapter 30, "Creating Form Fill Policies," on page 481.

  When you have finished the policy modifications, continue with Step 7.
- To create a new policy, click *Manage Policies*. On the Policies page, click *New*, specify a display name, select *Access Gateway: Form Fill* as the type, then click *OK*. For configuration information, see Chapter 30, "Creating Form Fill Policies," on page 481.

  When you have created your new policy, continue with Step 6.

6 To enable the policy you just created, select the policy, then click *Enable*.

Only the policies that are enabled are applied to this resource. All available Form Fill policies are listed. If you use the same policy for multiple protected resources, use the policy description field to indicate this.

7 To save your changes to browser cache, click *OK*.

8 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

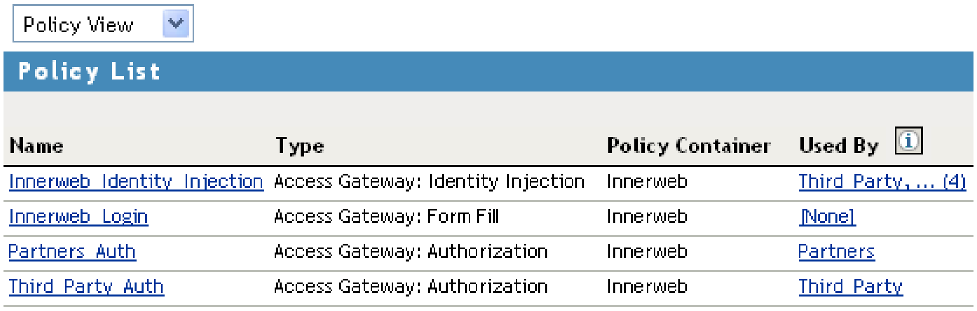
IMPORTANT: If you enable a Form Fill policy for a protected resource that has been assigned to use a contract that does not prompt the user for a password and the Form Fill policy contains a field for the user's password, single sign-on cannot be enabled because the password is not available. To enable single sign-on, you need to use an Authentication class that retrieves the user's password and injects it into the user's credentials when the user authenticates using a non-password method such as X509, Radius, smart card, or Kerberos. For information about such a class and how to download and configure it, see Access Management Authentication Class Extension to Retrieve Password for Single Sign-on (http://www.novell.com/communities/node/4556).

13.4.5 Assigning a Policy to Multiple Protected Resources

If you have created multiple protected resources that need to be protected by the same policy or policies, you can use the policy view to assign a policy to multiple protected resources. The one limitation is that the protected resources must belong to the same proxy service.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Reverse Proxy Name]* > *[Name of Proxy Service]* > *Protected Resources*.

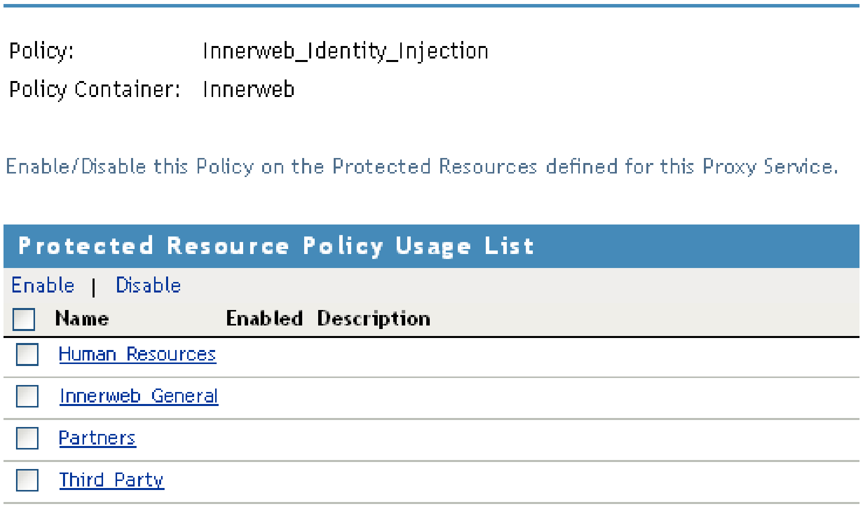
2 Select the *Policy View*.



Policy List

Name	Type	Policy Container	Used By
Innerweb_Identity_Injection	Access Gateway: Identity Injection	Innerweb	Third Party, ... (4)
Innerweb_Login	Access Gateway: Form Fill	Innerweb	[None]
Partners_Auth	Access Gateway: Authorization	Innerweb	Partners
Third_Party_Auth	Access Gateway: Authorization	Innerweb	Third Party

3 Select the *Used By* link of the policy you want to assign to multiple resources.


Policy: Innerweb_Identity_Injection
Policy Container: Innerweb

Enable/Disable this Policy on the Protected Resources defined for this Proxy Service.

Protected Resource Policy Usage List

☐	Name	Enabled	Description
☐	Human_Resources		
☐	Innerweb_General		
☐	Partners		
☐	Third_Party		

The *Policy* and *Policy Container* fields identify the policy. The *Protected Resource Policy Usage List* displays the protected resources defined for this proxy service and indicates which resources the policy has been enabled on.

4 To enable the policy for multiple resources, either select them one by one or click *Name* to select all of them, then click *Enable*. To disable a policy for a resource, select the resource, then click *Disable*.

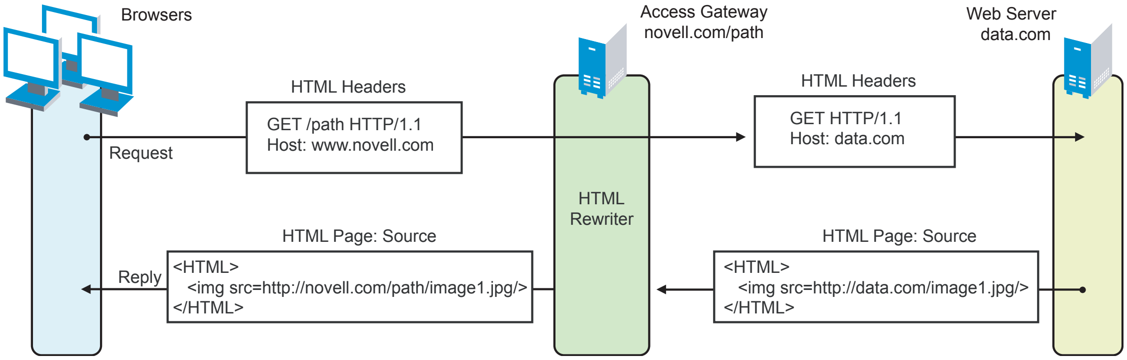
5 To save your changes to browser cache, click *OK*.

6 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.5 Configuring HTML Rewriting

Access Gateway configurations generally require HTML rewriting because the Web servers are not aware that the Access Gateway machine is obfuscating their DNS names. URLs contained in their pages must be checked to ensure that these references contain the DNS names that the client browser understands. On the other end, the client browsers are not aware that the Access Gateway is obfuscating the DNS names of the resources they are accessing. The URL requests coming from the client browsers that use published DNS names must be rewritten to the DNS names that the Web servers expect. Figure 13-3 illustrates these processes.

Figure 13-3 *HTML Rewriting*



The following sections describe the HTML rewriting process:

- Section 13.5.1, "Understanding the Rewriting Process," on page 216
- Section 13.5.2, "Specifying the DNS Names to Rewrite," on page 218
- Section 13.5.3, "Defining the Requirements for the Rewriter Profile," on page 220
- Section 13.5.4, "Configuring the HTML Rewriter and Profile," on page 227
- Section 13.5.5, "Disabling the Rewriter," on page 231

13.5.1 Understanding the Rewriting Process

The Access Gateway needs to rewrite URL references under the following conditions:

- To ensure that URL references contain the proper scheme (HTTP or HTTPS).

  If your Web servers and Access Gateway machines are behind a secure firewall, you might not require SSL sessions between them, and only require SSL between the client browser and the Access Gateway. For example, an HTML file being accessed through the Access Gateway for the Web site novell.com might have a URL reference to http://novell.com/path/image1.jpg. If the reverse proxy for novell.com/path is using SSL sessions between the browser and Access Gateway, the URL reference http://novell.com/path/image1.jpg must be rewritten to https://novell.com/path/image1.jpg. Otherwise, when the user clicks this link, the browser bounces between HTTP and HTTPS to establish a new SSL session.

- To ensure that URL references containing private IP addresses or private DNS names are changed to the published DNS name of the Access Gateway or hosts.

  For example, suppose that a company has an internal Web site named data.com, and wants to expose this site to Internet users through the Access Gateway using a published DNS name of novell.com. Many of the HTML pages on this Web site have URL references that contain the private DNS name, such as http://data.com/imagel.jpg. Because Internet users are unable to resolve data.com/imagel.jpg, links using this URL reference would return DNS errors in the browser.

  The HTML rewriter can resolve this issue. The DNS name field in the Access Gateway configuration is set to novell.com, which users can resolve through a public DNS server to the Access Gateway. The rewriter parses the Web page, and any URL references matching the private DNS name or private IP address listed in the Web server address field of the Access Gateway configuration are rewritten to the published DNS name novell.com and the port number of the Access Gateway.

Rewriting URL references addresses two issues: 1) URL references that are unreachable because of the use of private DNS names or IP addresses are now made accessible and 2) Rewriting prevents the exposure of private IP addresses and DNS names that might be sensitive information.

- To ensure that the Host header in incoming HTTP packets contains the name understood by the internal Web server.

  Using the example in Figure 13-3 on page 216, suppose that the internal Web server expects all HTTP or HTTPS requests to have the *Host* field set to data.com. When users send requests using the published DNS name novell.com/path, the *Host* field of the packets in those requests received by the Access Gateway is set to novell.com. The Access Gateway can be configured to rewrite this public name to the private name expected by the Web server by setting the *Web Server Host Name* option to data.com. Before the Access Gateway forwards packets to the Web server, the *Host* field is changed (rewritten) from novell.com to data.com. For information about configuring this option, see "Configuring the Web Servers of a Proxy Service" on page 205.

The rewriter searches for URLs in the following HTML contexts. They must meet the following criteria to be rewritten:

Context	**Criteria**
HTTP Headers	Qualified URL references occurring within certain types of HTTP response headers such as Location and Content-Location are rewritten. The Location header is used to redirect the browser to where the resource can be found. The Content-Location header is used to provide an alternate location where the resource can be found.
JavaScript	Within JavaScript*, absolute references are always evaluated for rewriting. Relative references (such as `index.html)` are not attempted. Absolute paths (such as `/docs/file.html`) are evaluated if the page is read from a path-based multi-homing Web server and the reference follows an HTML tag. For example, the string `href=‘/docs/file.html’` is rewritten if `/docs` is a multi-homing path that has been configured to be stripped.

HTML Tags	URL references occurring within the following HTML tag attributes are evaluated for rewriting:
	`action` `archive` `background`
	`base` `borderimage` `cite`
	`code` `codebase` `data`
	`dynscr` `href` `longdesc`
	`lowsrc` `onclick` `pluginspage`
	`src` `usemap`

References	An absolute reference is a reference that has all the information needed to locate a resource, including the hostname, such as `http://internal.web.site.com/index.html`. The rewriter always attempts to rewrite absolute references. The rewriter attempts to rewrite an absolute path when it is the multi-homing path of a path-based multi-homing service. For example, `/docs/file1.html` is rewritten if `/docs` is a multi-homing path that has been configured to be stripped. Relative references are not rewritten.

Context	Criteria
Query Strings	URL references contained within query strings can be configured for rewriting on path-based multi-homing proxy services.
Post Data	URL references specified in Post Data can be configured for rewriting on path-based multi-homing proxy services.

13.5.2 Specifying the DNS Names to Rewrite

The rewriter parses and searches the Web content that passes through the Access Gateway for URL references that qualify to be rewritten. URL references are rewritten when they meet the following conditions:

- URL references containing DNS names or IP addresses matching those in the Web server address list are rewritten with the *Published DNS Name*.
- URL references matching the *Web Server Host Name* are rewritten with the *Published DNS Name*.
- URL references matching entries in the *Additional DNS Name List* of the host are rewritten with the *Published DNS Name*. The *Web Server Host Name* does not need to be included in this list.
- The DNS names in the *Exclude DNS Name List* specify the names that the rewriter should skip and not rewrite.

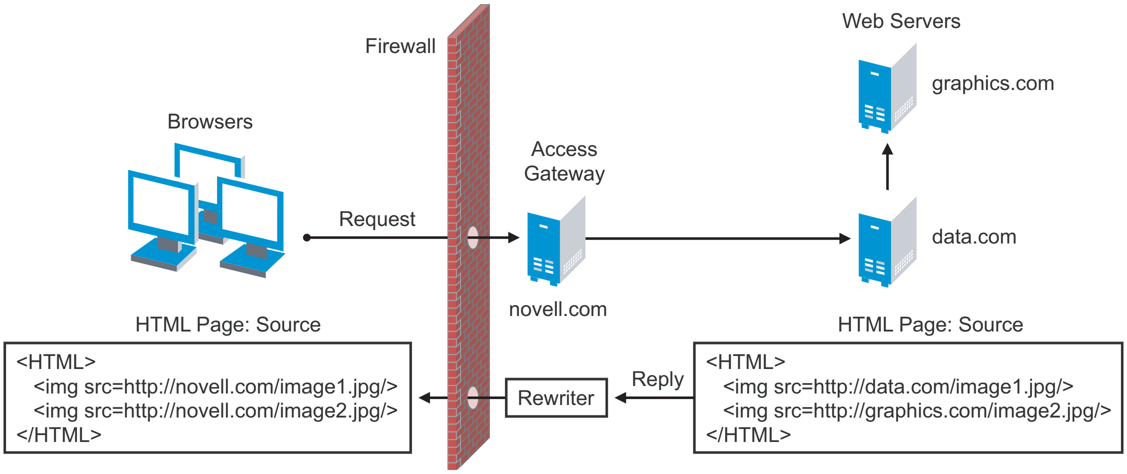
The following sections describe the conditions to consider when adding DNS names to the lists:

- “Determining Whether You Need to Specify Additional DNS Names” on page 218
- “Determining Whether You Need to Exclude DNS Names from Being Rewritten” on page 219

Determining Whether You Need to Specify Additional DNS Names

Sometimes Web pages contain URL references to a host name that does not meet the default criteria for being rewritten. That is, the URL reference does not match the *Web Server Host Name* or any value (IP address) in the *Web Server List*. If these names are sent back to the client, they are not resolvable. Figure 13-4 illustrates a scenario that requires an entry in the *Additional DNS Name List*.

Figure 13-4 *Rewriting a URLs for Web Servers*


Firewall
Web Servers
graphics.com
Browsers
Access Gateway
Request
data.com
novell.com
HTML Page: Source
HTML Page: Source
<HTML>
<img src=http://novell.com/image1.jpg/>
<img src=http://novell.com/image2.jpg/>
</HTML>
Rewriter
Reply
<HTML>
<img src=http://data.com/image1.jpg/>
<img src=http://graphics.com/image2.jpg/>
</HTML>

The page on the data.com Web server contains two links, one to an image on the data.com server and one to an image on the graphics.com server. The link to the data.com server is automatically rewritten to novell.com, when rewriting is enabled. The link to the image on graphics.com is not rewritten, until you add this URL to the *Additional DNS Name List*. When the link is rewritten, the browser knows how to request it, and the Access Gateway knows how to resolve it.

You need to include names in this list if your Web servers have the following configurations:

- If you have a cluster of Web servers that are not sharing the same DNS name, you need to add their DNS names to this list.
- If your Web server obtains content from another Web server, the DNS name to this additional Web server needs to rewritten.
- If the Web server listens on one port (for example, 80), and redirects the request to a secure port (for example, 443). The response to the user comes back on https://<*DNS_name*>:443. This does not match the request which was sent on http://<*DNS_name*>:80. If you add the DNS name to the list, the response can be sent in the format that the user expects.
- If an application is written to use a private host name. For example, assume that an application URL reference contains the host name of home (`http://home/index.html`). This host name would need to be added to the *Additional DNS Name List*.
- If you enable the *Forward Received Host Name* option on your path-based multi-homing service and your Web server is configured to use a different port, you need to add the DNS name with the port to the *Additional DNS Name List*.

  For example, if the public DNS name of the proxy service is www.mylag.com, the path for the path-based multi-homing service is /sales, and the Web server port is 801, the following DNS name needs to be added to the *Additional DNS Name List* of the /sales service:

  ```
  http://www.mylag.com:801
  ```

When you enter a name in the list, it can use any of the following formats:

```
DNS_name
host_name
IP_address
scheme://DNS_name
scheme://IP_address
scheme://DNS_name:port
scheme://IP_address:port
```

For example:

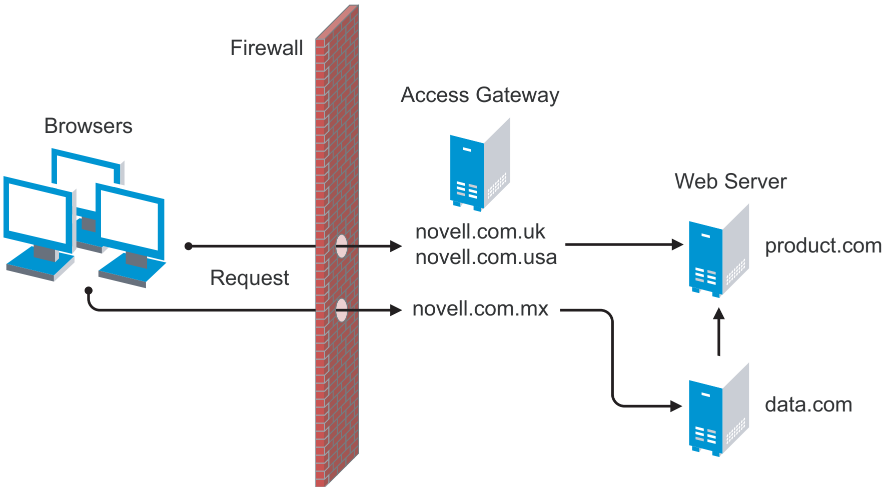
```
HOME
https://www.backend.com
https://10.10.15.206:444
```

These entries are not case sensitive.

Determining Whether You Need to Exclude DNS Names from Being Rewritten

If you have two reverse proxies protecting the same Web server, the rewriter correctly rewrites the references to the Web server so that browser always uses the same reverse proxy. In other words, if the browser requests a resource using acme.com.uk, the response is returned with references to acme.com.uk and not acme.com.usa. If you have a third reverse proxy protecting a Web server, the rewriting rules can become ambiguous. For example, consider the configuration illustrated in Figure 13-5.

Figure 13-5 Excluding URLs



A user accesses data.com through the published DNS name of novell.com.mx. The data.com server has references to product.com. The novell.com.mx proxy has two ways to get to the product.com server because this Web server has two published DNS names (novell.com.uk and novell.com.usa). The rewriter could use either of these names to rewrite references to product.com.

- If you want all users coming through novell.com.mx to use the novell.com.usa proxy, you need to block the rewriting of product.com to novell.com.uk. On the HTLM Rewriting page of the reverse proxy for novell.com.uk, add product.com and any aliases to the *Exclude DNS Name List*.
- If you do not care which proxy is returned in the reference, you do not need to add anything to the *Exclude DNS Names List*.

13.5.3 Defining the Requirements for the Rewriter Profile

An HTML rewriter profile allows you to customize the rewriting process and specify which profile is selected to rewrite content on a page. This section describes the following features of the rewriter profile:

- "Types of Rewriter Profiles" on page 220
- "Page Matching Criteria for Rewriter Profiles" on page 221
- "Possible Actions for Rewriter Profiles" on page 222
- "String Replacement Rules for Word Profiles" on page 224
- "String Replacement Rules for Character Profiles" on page 225
- "Using $path to Rewrite Paths in JavaScript Methods, Parameters, or Variables" on page 226

Types of Rewriter Profiles

The Access Gateway allows you to define two types of profiles:

- "Word Profile" on page 221
- "Character Profile" on page 221

Word Profile

A Word profile searches for matches on words. For example, “get” matches the word “get” and any word that begins with “get” such as “getaway” but it does not match the “get” in “together” or “beget.”

The Access Gateway has a default Word profile. It is not specific to a reverse proxy or its proxy services. When you modify its behavior, remember its scope.

If you enable HTML rewriting, but do not define a Word profile for the proxy service, the default Word profile is used. This profile is preconfigured to rewrite the *Web Server Host Name* and any other names listed in the *Additional DNS Name List*. The preconfigured profile matches all URLs with the following content-types:

text/html	text/javascript
text/xml	application/javascript
text/css	application/x-javascript

If this default behavior does not match your requirements for a particular page, create your own Word profile and position it before the default profile in the list of profiles. Only one Word profile is applied per page. The first Word profile that matches the page is applied. Profiles lower in the list are ignored.

For information about how strings are replaces in a Word profile, see the following:

- “String Replacement Rules for Word Profiles” on page 224
- “Using $path to Rewrite Paths in JavaScript Methods, Parameters, or Variables” on page 226

Character Profile

A Character profile searches for matches on a specified set of characters. For example, “top” matches the word “top” and the “top” in “tabletop,” “stopwatch,” and “topic.”

If need functionality not provided by the default profile, create a Character profile. If you create multiple Character profiles, order is important. The first Character profile that matches the page is applied. Profiles lower in the list are ignored.

For information on how strings are replaced in a Character profile, see “String Replacement Rules for Character Profiles” on page 225.

Page Matching Criteria for Rewriter Profiles

You specify the following matching criteria for selecting the profile:

- The URLs to match
- The URLs that cannot match
- The content types to match

You use the *Requested URLs to Search* section of the profile to set up the matching policy.

URLs: The URLs specified in the policy should use the following formats:

Sample URL	Description
http://www.a.com/content	Matches pages only if the request URL does not contain a trailing slash.
http://www.a.com/content/	Matches pages only if the request URL does contain a trailing slash.
http://www.a.com/content/index.html	Matches only this specific file.
http://www.a.com/content/*	Matches the request URL whether or not it has a trailing slash and matches all files in the directory.
http://www.a.com/*	Matches the proxy service and everything it is protecting.

You can specify two types of URLs. In the *If Requested URL Is* list, you specify the URLs of the pages you want this profile to match. In the *And Requested URL Is Not* list, you specify the URLs you don't want this profile to match. You can use the asterisk wildcard for a URL in the *If Requested URL Is* list that matches pages you really don't want this profile to match, then use a URL in the *And Requested URL Is Not* list to exclude them from matching. If a page matches both a URL in the *If Requested URL Is* list and in the *And Requested URL Is Not* list, the profile does not match the page.

For example, you could specify the following URL in the *If Requested URL Is* list:

```
http://www.a.com/*
```

You could then specify the following URL in the *And Requested URL Is Not* list:

```
http://www.a.com/content/*
```

These two entries cause the profile to match all pages on the www.a.com Web server except for the pages in the `/content` directory and its subdirectories.

IMPORTANT: If nothing is specified in either of the two lists, the profile skips the URL matching requirements and uses the content-type to determine if a page matches.

Content-Type: In the *And Document Content-Type Is* section, you specify the content-types you want this profile to match. To add a new content-type, click *New* and specify the name such as `text/dns`. Search your Web pages for content-types to determine if you need to add new types. To add multiple values, enter each value on a separate line.

Regardless of content-type, the page matches if the file extension is `html, htm, shtml, jhtml, asp,` or `jsp`.

Possible Actions for Rewriter Profiles

The rewriter action section of the profile determines the actions the rewriter performs when a page matches the profile. Select from the following:

- Strip Path Actions
- Enabling or Disabling Rewriting
- Replacing URLs in JavaScript Variables and HTML Attributes
- Replacing URLs in Java Methods

- String Replacement

Strip Path Actions: A profile might require the strip path options if the proxy service has the following characteristics:

- It is a path-based multi-homing proxy.
- The *Remove Path on Fill* option has been enabled.
- URLs appear in query strings or Post Data.

If your profile needs to match pages from this type of proxy server, you might need to enable the *Strip Path from Query String* and *Strip Path from Post Data* options.

The strip path options are not available for a Character profile. If the proxy service is not a path-based multi-homing proxy, the strip path options have no effect.

Enabling or Disabling Rewriting: The *Enable Rewriter Actions* option determines whether the rewriter performs any actions:

- Select the option to have the rewriter rewrite the references and data on the page.
- Leave the option unselected to disable rewriting. This allows you to create a profile for the pages you do not want rewritten.

Replacing URLs in JavaScript Variables and HTML Attributes: The *Variable and Attribute Name* list allows you to specify the HTML attributes or JavaScript variables that you want searched for DNS names that might need to be rewritten. For the list of HTML attribute names that are automatically searched, see "HTML Tags" on page 217. You might want to add the following attributes:

- **value:** This attribute enables the rewriter to search the `<param>` elements on the HTML page for value attributes and rewrite the value attributes that are URL strings.

  If you need more granular control (some need to be rewritten but others do not) and you can modify the page, see "Disabling with Page Modifications" on page 232.
- **formvalue:** This attribute enables the rewriter to search the `<form>` element on the HTML page for `<input>`, `<button>`, and `<option>` elements and rewrite the value attributes that are URL strings. For example, if your multi-homing path is `/test` and the form line is `<input name="navUrl" type="hidden" value="/IDM/portal/cn/GuestContainerPage/656gwmail">`, this line would be rewritten to the following value before sending the response to the client:

  ```
  <input name="navUrl" type="hidden" value="/test/IDM/portal/cn/
  GuestContainerPage/656gwmail">
  ```

  The formvalue attribute enables the rewriting of all URLs in the `<input>`, `<button>`, and `<option>` elements in the form. If you need more granular control (some need to be rewritten but others do not) and you can modify the form page, see "Disabling with Page Modifications" on page 232.

This option is not available for a Character profile.

Replacing URLs in Java Methods: The *And JavaScript Method to Search for Is* list allows you to specify the Java methods to search to see if their parameters contain a URL string.

This option is not available for a Character profile.

String Replacement: The *Additional Strings to Replace* list allows you to search for a string and replace it.

When defining a rewriter profile, you should try to put all the string actions into the Word profile. When a Word profile and a Character profile both match the same URL, you need to ensure that they do not contain search and replace actions for overlapping strings. The results of such actions are unpredictable.

For example, if your Word profile has an action to search for `Doodle` and replace this string with `Artwork` and your Character profile has an action to search for `Doo` and replace this string with `Zoo`, the results are unpredictable. If you place both search and replace actions in the Word profile, the results are predictable.

For the rules and tokens that can be used in the search strings, see the following:

- "String Replacement Rules for Word Profiles" on page 224
- "String Replacement Rules for Character Profiles" on page 225

For information on how the *Additional Strings to Replace* list can be used to reduce the number of Java methods you need to list, see "Using $path to Rewrite Paths in JavaScript Methods, Parameters, or Variables" on page 226.

String Replacement Rules for Word Profiles

In a Word profile, a string matches all paths that start with the characters in the specified string. For example:

Search String	Matches This String	Doesn't Match This String

`/path`	`/path`
	`/pathother`
	`/path/other`
	`/path.html`
		`/mypath`

You can use the following special tokens to modify the default matching rules:

- [w] to match one white space character
- [ow] to match 0 or more white space characters
- [ep] to match a path element in a URL path, excluding words that end in a period
- [ew] to match a word element in a URL path, including words that end in a period
- [oa] to match one or more alphanumeric characters

White Space Tokens: You use the [w] and the [ow] tokens to specify where white space might occur in the string. For example:

```
[ow]my[w]string[w]to[w]replace[ow]
```

If you don't know, or don't care, whether the string has zero or more white characters at the beginning and at the end, use [ow] to specify this. The [w] specifies exactly one white character.

Path Tokens: You use the [ep] and [ew] tokens to match path strings. The [ep] token can be used to match the following types of paths:

Search String	Matches This String	Doesn't Match This String
`/path[ep]`	`/path`	`/path.html`
	/home/path/other	`/home/pathother`

The [ew] token can be used to match the following types of paths:

Search String	Matches This String	Doesn't Match This String
`/path[ew]`	`/path.html`	/paths
	`/home/path`	

Name Tokens: You use the [oa] token to match function or parameter names that have a set string to start the name and end the name, but the middle part of the name is a computer-generated alphanumeric string. For example, the [oa] token can be used to match the following types of names:

Search String	Matches This String	Doesn't Match This String
javaFunction-[oa](	javaFunction-1234a56()	javaFunction()
	javaFunction-a()	

String Replacement Rules for Character Profiles

When you configure multiple strings for replacement, the rewriter uses the following rules for determining how characters are replaced in strings:

- String replacement is done as a single pass.
- String replacement is not performed recursively. Suppose you have listed the following search and replacement strings:

  ```
  DOG     to be replaced with     CAT
  A       to be replaced with     O
  ```

  All occurrences of the string DOG are replaced with CAT, regardless of whether it is the word DOG or the word DOGMA. Only one replacement pass occurs. The rewritten CAT is not replaced with COT.
- Because string replacement is done in one pass, the string that matches first takes precedence. Suppose you have listed the following search and replacement strings:

  ```
  ABC       to be replaced with     XYZ
  BCDEF     to be replaced with     PQRSTUVWXYZ
  ```

  If the original string is ABCDEFGH, the replaced string is XYZDEFGH.
- If two specified search strings match the data portion, the search string of longer length is used for the replacement except for the case detailed above. Suppose you have listed the following search and replacement strings:

  ```
  ABC        to be replaced with     XYZ
  ABCDEF     to be replaced with     PQRSTUVWXYZ
  ```

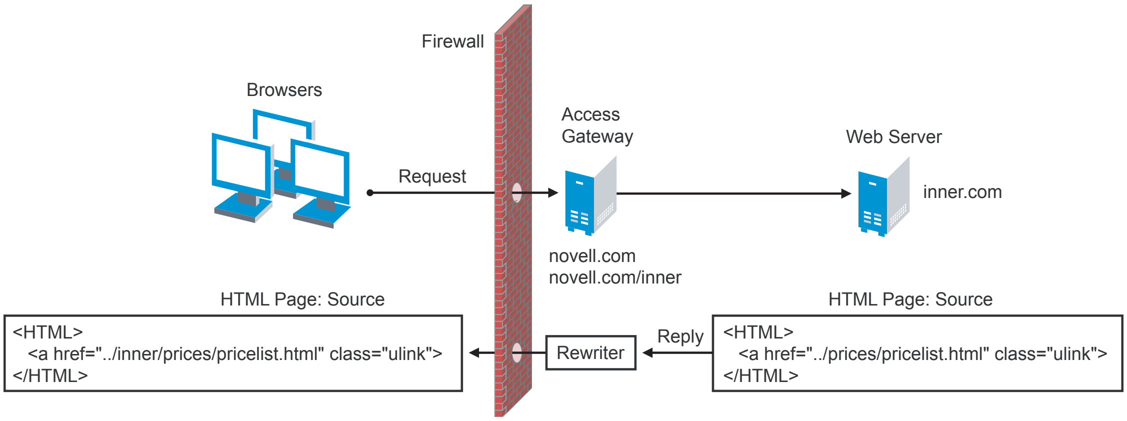
  If the original string is ABCDEFGH, the replaced string is PQRSTUVWXYZGH.

Using $path to Rewrite Paths in JavaScript Methods, Parameters, or Variables

You can use the $path token to rewrite paths on a path-based multi-homing service that has the *Remove Path on Fill* option enabled. This token is useful for Web applications that require a dedicated Web server and are therefore installed in the root directory of the Web server. If you protect this type of application with Access Manager using a path-based multi-homing proxy service, your clients access the application with a URL that contains a /path value. The proxy service uses the path to determine which Web server a request is sent to, and the path must be removed from the URL before sending the request to the Web server.

The application responds to the requests. If it uses JavaScript methods, parameters, or variables to generate paths to resources, these paths are sent to client without prepending the path for the proxy service. When the client tries to access the resource specified by the Web server path, the proxy service cannot locate the resource because the multi-homing path is missing. The figure below illustrates this flow with the rewriter adding the multi-homing path in the reply.

Figure 13-6 *Rewriting with a Multi-homing Path*



To make sure all the paths generated by JavaScript are rewritten, you must search the Web pages of the application. You can then either list all the JavaScript methods, parameters, and variables in the *Additional Names to Search for URL Strings to Rewrite with Host Name* section of the rewriter profile, or you can use the $path token in the *Additional Strings to Replace* section. This token, which is a shortcut for the multi-homing path, together with the *Strip Path from Query String* and *Strip Path from Post Data* actions, usually can find all the paths that need to rewritten. If nothing else, it reduces the number of JavaScript methods, parameters, and variables that you otherwise need to list individually.

To use the $path token, you add a search string and a replace string that uses the token. For example, if the /prices/pricelist.html page is generated by JavaScript and the multi-homing path for the proxy service is /inner, you would specify the following stings:

Table 13-1 *Search and Replace Strings*

Search String	Replacement String
/prices	$path/prices

This configuration allows the following paths to be rewritten.

Table 13-2 *Rewriting Strings Sent from the Web Server to the Browser*

Web Server String	Rewritten String for the Browser
/prices/pricelist.html	/inner/prices/pricelist.html
/prices	/inner/prices

If the *Strip Path from Query String* or *Strip Path from Post Data* option is enabled, the search and replace strings allow the following paths to be rewritten.

Table 13-3 *Rewriting Strings Sent from the Browser to the Web Server*

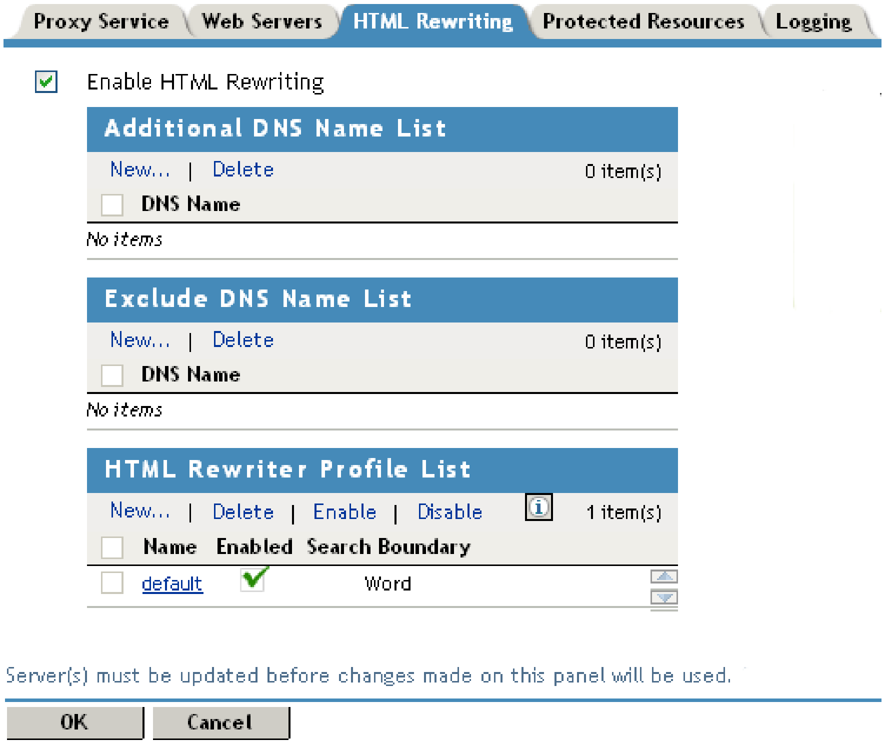
Browser String	Rewritten String for the Web Server
/inner/prices/pricelist.html	/prices/pricelist.html
/inner/prices	/prices

13.5.4 Configuring the HTML Rewriter and Profile

You configure the HTML rewriter for a proxy service, and these values are applied to all Web servers that are protected by this proxy service.

To configure the HTML rewriter:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *HTML Rewriting*.



☑ Enable HTML Rewriting

Additional DNS Name List

0 item(s)

☐	DNS Name

No items

Exclude DNS Name List

0 item(s)

☐	DNS Name

No items

HTML Rewriter Profile List

1 item(s)

☐	Name	Enabled	Search Boundary
☐	default	✓	Word

Server(s) must be updated before changes made on this panel will be used.

The HTML Rewriting page specifies which DNS names are to be rewritten. The HTML Rewriter Profile specifies which pages to search for DNS names that need to be rewritten.

2 Select *Enable HTML Rewriting*.

This option is enabled by default. When it is disabled, no rewriting occurs.When enabled, this option activates the internal HTML rewriter. This rewriter replaces the name of the Web server with the published DNS name when sending data to the browsers. It replaces the published DNS name with the *Web Server Host Name* when sending data to the Web server. It also makes sure the proper scheme (HTTP or HTTPS) is included in the URL. This is needed because you can configure the Access Gateway to use HTTPS between itself and client browsers and to use HTTP between itself and the Web servers.

3 In the *Additional DNS Name List* section, click *New*, specify a DNS that appears on the Web pages of your server (for example a DNS name other than the Web server's DNS name), then click *OK*.

For more information, see "Determining Whether You Need to Specify Additional DNS Names" on page 218.

4 In the *Exclude DNS Name List* section, click *New*, specify a DNS name that appears on the Web pages of your server that you do not want rewritten, then click *OK*.

For more information, see "Determining Whether You Need to Exclude DNS Names from Being Rewritten" on page 219.

5 Use the *HTML Rewriter Profile List* to configure a profile. Select one of the following actions:

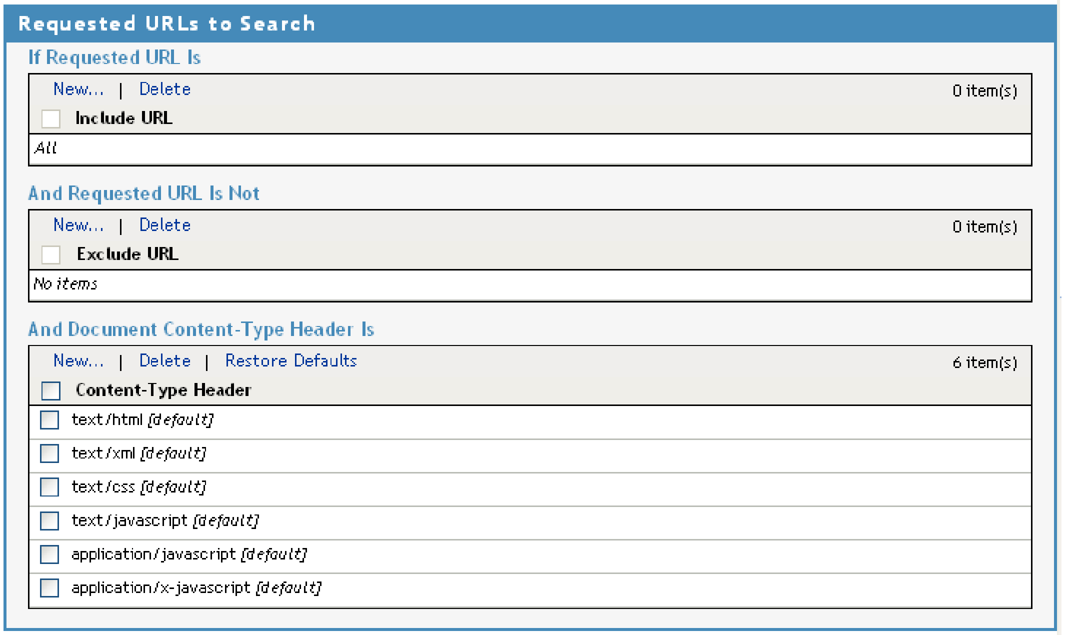
- **New:** To create a profile, click *New*. Specify a display name for the profile and select either a *Word* or *Character* for the *Search Boundary*. Continue with Step 6.
  - **Word:** A Word profile searches for matches on words. For example, "get" matches the word "get" and any word that begins with "get" such as "getaway" but it does not match the "get" in "together" or "beget."

    If you create multiple Word profiles, order is important. The first Word profile that matches the page is executed. Profiles lower in the list are ignored.
  - **Character:** A Character profile searches for matches on a specified set of characters. For example, "top" matches the word "top" and the "top" in "tabletop," "stopwatch," and "topic."

    If you want to add functionality to the default profile, create a Character profile. It has all the functionality of a Word profile, except searching for attribute names and Java variables and methods. If you create multiple Character profiles, order is important. The first Character profile that matches the page is executed. Profiles lower in the list are ignored.
- **Delete:** To delete a profile, select the profile, then click *Delete*. Continue with Step 13.
- **Enable:** To enable a profile, select the profile, then click *Enable*. Continue with Step 13.
- **Disable:** To disable a profile, select the profile, then click *Disable*. Continue with Step 13.
- **Modify:** To view or modify the current configuration for a profile, click the name of the profile. Continue with Step 6.

  The default profile is designed to be applied to all pages protected by the Access Gateway. It is not specific to a reverse proxy or its proxy services. If you modify its behavior, remember its scope. Rather than modify the default profile, you should create your own customized Word profile and enable it

6 Use the *Requested URLs to Search* section to set up a policy for specifying the URLs you want this profile to match.



Requested URLs to Search

If Requested URL Is

New... | Delete 0 item(s)

☐ **Include URL**

All

And Requested URL Is Not

New... | Delete 0 item(s)

☐ **Exclude URL**

No items

And Document Content-Type Header Is

New... | Delete | Restore Defaults 6 item(s)

☐ **Content-Type Header**

☐ text/html *[default]*

☐ text/xml *[default]*

☐ text/css *[default]*

☐ text/javascript *[default]*

☐ application/javascript *[default]*

☐ application/x-javascript *[default]*

Fill in the following fields:

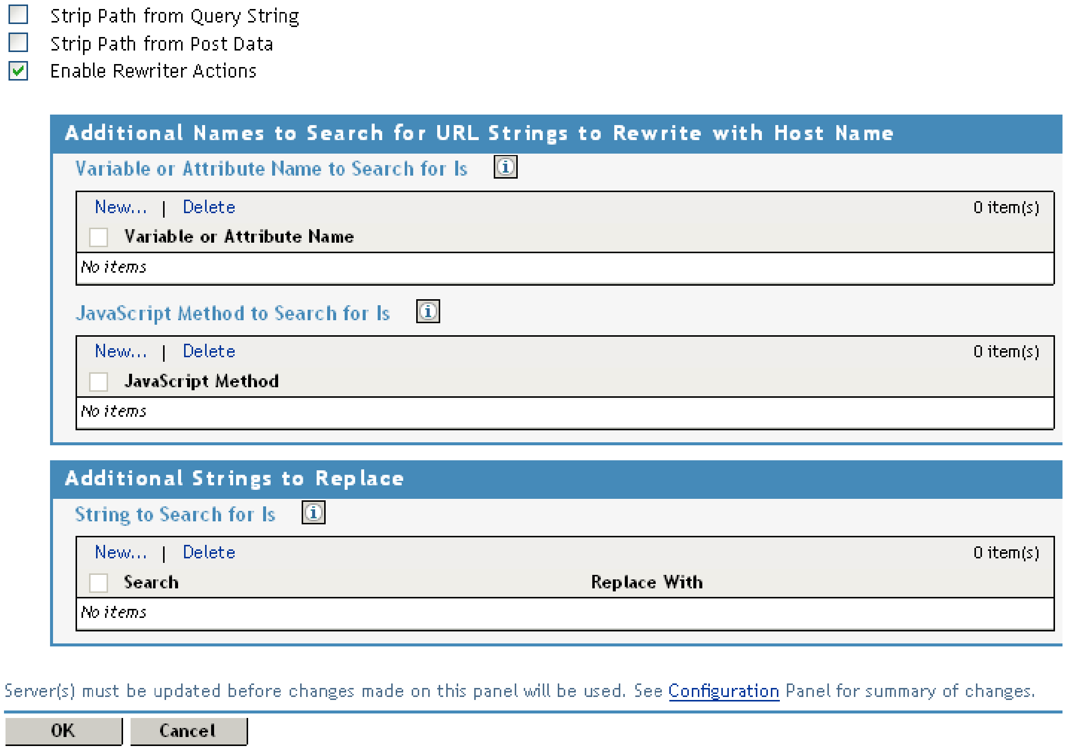
If Requested URL Is: Specify the URLs of the pages you want this profile to match. Click *New* to add a URL to the text box. To add multiple values, enter each value on a separate line.

And Requested URL Is Not: Specify the URLs of pages that this profile should not match. If a page matches the URL in both the *If Requested URL Is* list and *And Requested URL Is Not* list, profile does not match the page. Click *New* to add a URL to the text box. To add multiple values, enter each value on a separate line.

And Document Content-Type Is: Select the content-types you want this profile to match. To add a new content-type, click *New* and specify the name such as `text/dns`. Search your Web pages for content-types to determine if you need to add new types. To add multiple values, enter each value on a separate line.

For more information on how to use these options, see "Page Matching Criteria for Rewriter Profiles" on page 221.

7 Use the Actions section to specify the actions the rewriter should perform if the page matches the criteria in the *Requested URLs to Search* section.



☐ Strip Path from Query String
☐ Strip Path from Post Data
☑ Enable Rewriter Actions

Additional Names to Search for URL Strings to Rewrite with Host Name
Variable or Attribute Name to Search for Is
New... | Delete
0 item(s)
Variable or Attribute Name
No items
JavaScript Method to Search for Is
New... | Delete
0 item(s)
JavaScript Method
No items
Additional Strings to Replace
String to Search for Is
New... | Delete
0 item(s)
Search
Replace With
No items
OK
Cancel

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

Configure the following actions:

Strip Path from Query String: (Not available for Character profiles) Select this option to remove the path from the query string. To use this option, your proxy service must meet the conditions listed in "Possible Actions for Rewriter Profiles" on page 222.

Strip Path from Post Data: (Not available for Character profiles) Select this option to remove the path from the Post Data command. To use this option, your proxy service must meet the conditions listed in "Possible Actions for Rewriter Profiles" on page 222.

Enable Rewriter Actions: Select this action to enable the rewriter to perform any actions:

- Select it to have the rewriter use the profile to rewrite references and data on the page. If this option is not selected, you cannot configure the action options.
- Leave it unselected to disable rewriting. This allows you to create a profile for the pages you do not want rewritten.

8 (Not available for Character profiles) If your pages contain JavaScript, use the *Additional Names to Search for URL Strings to Rewrite with Host Name* section to specify JavaScript variables or methods. You can also add HTML attribute names. (For the list of attribute names that are automatically searched, see "HTML Tags" on page 217.)

Fill in the following fields:

Variable or Attribute Name to Search for Is: Lists the name of an HTML attribute or JavaScript variable to search to see if its value contains a URL string. Click *New* to add a name to the text box. To add multiple values, enter each value on a separate line.

JavaScript Method to Search for Is: Lists the names of Java methods to search to see if their parameters contain a URL string. Click *New* to add a method to the text box. To add multiple values, enter each value on a separate line.

9. Use the *Additional Strings to Replace* section to specify a string to search for and specify the text it should be replaced with. The search boundary (word or character) that you specified when creating the profile is used when searching for the string.

   To add a string, click *New*, then fill in the following:

   Search: Specify the string you want to search for. The profile type controls the matching and replacement rules. For more information, see one of the following:

   - "String Replacement Rules for Character Profiles" on page 225
   - "String Replacement Rules for Word Profiles" on page 224
   - "Using $path to Rewrite Paths in JavaScript Methods, Parameters, or Variables" on page 226

   Replace With: Specify the string you want to use in place of the search string.

10. Click *OK*.

11. If you have more than one profile in the *HTML Rewriter Profile List*, use the up-arrow and down-arrow buttons to order the profiles.

    If you create more than one profile, order becomes important. For example if you want to rewrite all pages with a general rewriter profile (with a URL such as /*) and one specific set of pages with another rewriter profile (with a URL such as /doc/100506/*), you need to have the specific rewriter profile listed before the general rewriter profile. Only one Word profile and one Character profile are executed per page. The first one in the list that matches a page is executed, and the others are ignored.

12. Enable the profiles you want to use for this protected resource. Select the profile, then click *Enable*.

    The default profile cannot be disabled. However, it is not executed if you have enabled another Word profile that matches your pages, and this profile comes before the default profile in the list.

13. To save your changes to browser cache, click *OK*.

14. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

15. The cached pages affected by the rewriter changes must be updated on the Access Gateway. Do one of the following:

    - If the changes affect numerous pages, click *Access Gateways*, select the name of the server, then click *Actions* > *Purge All Cache*.
    - If the changes affect only a few pages, you can update them from a browser. Access the page, then press Ctrl+Shift+Refresh to force a refresh of the page.

13.5.5 Disabling the Rewriter

There are three methods you can use to disable the internal rewriter:

- "Disabling per Proxy Service" on page 232
- "Disabling per URL" on page 232
- "Disabling with Page Modifications" on page 232

Disabling per Proxy Service

By default, the rewriter is enabled for all proxy services. The rewriter can slow performance because of the parsing overhead. In some cases, a Web site might not have content with URL references that need to be rewritten. The rewriter can be disabled on the proxy service that protects that Web site.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *HTML Rewriting*.
2. Deselect the *Enable HTML Rewriting* option, then click *OK*.
3. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.
4. Select the Access Gateway, then click *Actions* > *Purge All Cache* > *OK*.

Disabling per URL

You can also specify a list of URLs that are to be excluded from being rewritten for the selected proxy service.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *HTML Rewriting*.
2. Click the name of the Word profile defined for this proxy service.

   If you have not defined a custom Word profile for the proxy service, you might want to create one. If you modify the default profile, those changes are applied to all proxy services.
3. In the *And Requested URL Is Not* section, click *New*, then specify the names of the URLs you do not want rewritten.

   Specify each URL on a separate line.
4. Click *OK* twice
5. In the *HTML Rewriter Profile List*, make sure the profile you have modified is enabled and at the top of the list, then click *OK*.
6. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.
7. Select the Access Gateway, then click *Actions* > *Purge All Cache* > *OK*.

Disabling with Page Modifications

There are cases when the URLs in only part of a page or in some of the JavaScript or form can be rewritten and the rest should not be rewritten. When this is the case, you might need to modify the content on the Web server. Although this deviates from the design behind Access Manager, you might encounter circumstances where it cannot be avoided.

You can add the following types of tags to the pages on the Web server:

- Page Tags
- Param Tags
- Form Tags

These tags are seen by browsers as a comment mark, and do not show up on the screen (except possibly on older browser versions).

NOTE: If the pages you modify are cached on the Access Gateway, you need to purge the cache before the changes become effective.

Page Tags: In the case where you want only portions of a page rewritten, you can add the following tags to the page.

```
<!--NOVELL_REWRITER_OFF-->
.
.
HTML data not to be rewritten
.
.
<!--NOVELL_REWRITER_ON-->
```

The last tag is optional, and if omitted, it prevents the rest of the page from being rewritten after the initial tag is encountered.

Param Tags: Sometimes the JavaScript on the page contains `<param>` elements that contain a value attribute with a URL. You can enable global rewriting of this attribute by adding `value` to the list of variable and attribute names to search for. If you need more control because some URLs need to be rewritten but others cannot be rewritten, you can turn on and turn off the `value` rewriting by adding the following tags before and after the `<param>` element in the JavaScript.

```
<!--NOVELL_REWRITE_ATTRIBUTE_ON='value'-->
.
.
<param> elements to be rewritten
.
.
<!--NOVELL_REWRITE_ATTRIBUTE_OFF='value'-->
.
.
<param> elements that shouldn’t be rewritten
```

Form Tags: Some applications have forms in which the `<input>`, `<button>`, and `<option>` elements contain a value attribute with a URL. You can enable global rewriting of these attributes by adding `formvalue` to the list of variable and attribute names to search for. If you need more control because some URLs need to be rewritten but others cannot be rewritten, you can turn on and turn off the `formvalue` rewriting by adding the following tags before and after the `<input>`, `<button>`, and `<option>` elements in the form.

```
<!--NOVELL_REWRITE_ATTRIBUTE_ON='formvalue'-->
.
.
<input>, <button>, and <option> elements to be rewritten
.
.
<!--NOVELL_REWRITE_ATTRIBUTE_OFF='formvalue'-->
.
.
<input>, <button>, and <option> elements that shouldn’t be rewritten
```

13.6 Configuring Connection and Session Limits

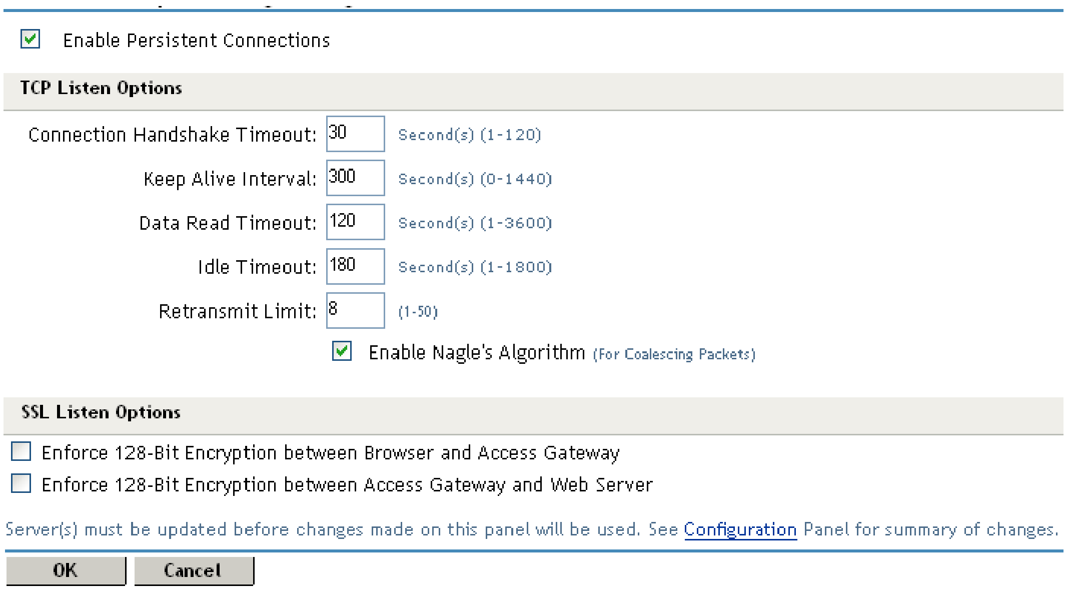
The Access Gateway establishes connections with clients and with Web servers. The Identity Server establishes the session and sets the session timeout. For most networks, the default values for the connection and session limits provide adequate performance, but you can fine-tune the options to match for your network, its performance requirements, and your users:

- Section 13.6.1, “Configuring TCP Listen Options for Clients,” on page 234
- Section 13.6.2, “Configuring TCP Connect Options for Web Servers,” on page 235
- Section 13.6.3, “Configuring Connection and Session Persistence,” on page 237
- Section 13.6.4, “Configuring the Session Timeout,” on page 238

13.6.1 Configuring TCP Listen Options for Clients

The TCP listen options allow you to control how idle and unresponsive browser connections are handled and to optimize these processes for your network. For most networks, the default values provide adequate performance. If your network is congested and slow, you might want to increase some of the limits.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *TCP Listen Options*.



☑ Enable Persistent Connections

TCP Listen Options

Connection Handshake Timeout: 30 Second(s) (1-120)

Keep Alive Interval: 300 Second(s) (0-1440)

Data Read Timeout: 120 Second(s) (1-3600)

Idle Timeout: 180 Second(s) (1-1800)

Retransmit Limit: 8 (1-50)

☑ Enable Nagle's Algorithm (For Coalescing Packets)

SSL Listen Options

☐ Enforce 128-Bit Encryption between Browser and Access Gateway

☐ Enforce 128-Bit Encryption between Access Gateway and Web Server

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2 Select *Enable Persistent Connections* to allow the Access Gateway to establish a persistent HTTP connection between the Access Gateway and the browser. Usually, HTTP connections service only one request and response sequence. A persistent connection allows multiple requests to be serviced before the connection is closed.

This option is enabled by default.

3 Specify values for the following fields:

Connection Handshake Timeout: Sets a timeout limit for a connecting device that stops responding after having initiated the TCP handshake process. If an expected handshake response is not received from the connecting device in this amount of time, an error occurs. Setting the value lower might help defend against SYN attacks. The timeout can be set from 1 to 120 seconds. The default is 30 seconds.

Keep Alive Interval: Sets the length of time between packets being sent to a connected device to determine if the connection is still alive. If a response is not received within the Data Read

Timeout value, the connection is closed. On an idle connection, sending these ping packets continues until the Idle Timeout value is reached. Setting the value to zero prevents the sending of keep-alive packets. The value can be set from 0 to 1440 seconds (24 minutes). The default is 300 seconds (5 minutes).

Data Read Timeout: Determines when an unresponsive connection is closed. When exchanging data, if an expected response from the connected device is not received within this amount of time, the connection is closed. This value might need to be increased for slow or congested network links. The value can be set from 1 to 3600 seconds (1 hour). The default is 120 seconds (2 minutes).

Idle Timeout: Determines when an idle connection is closed. If no application data is exchanged over a connection for this amount of time, the connection is closed. This value limits how long an idle persistent connection is kept open. This setting is a compromise between freeing resources to allow additional inbound connections, and keeping connections established so that new connections from the same device do not need to be re-established. The value can be set from 1 to 1800 seconds (30 minutes). The default is 180 seconds (3 minutes).

Retransmit Limit: Determines how many times data is resent. When exchanging data, if the expected acknowledgement (ACK) response is not received, this is the number of times the device attempts to resend the data before closing the connection. You can set the value from 1 - 50. The default is 8.

Enable Nagle's Algorithm: Determines whether small buffer messages can be concatenated into one large message. When this option is enabled, small buffer messages are automatically concatenated. This process increases the efficiency of a network application system by decreasing the number of packets that must be sent. Enabling this feature delays data transmission until a full TCP packet can be sent.

4. On a Linux Access Gateway, you can also configure the encryption key. (For the NetWare® Access Gateway, the encryption key is set globally for all reverse proxies. See Section 14.6, "Configuring the Encryption Key," on page 248.) Select one or more of the following:

   Enforce 128-Bit Encryption between Browser and Access Gateway: When this option is selected, the Access Gateway requires all its server connections with client browsers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.

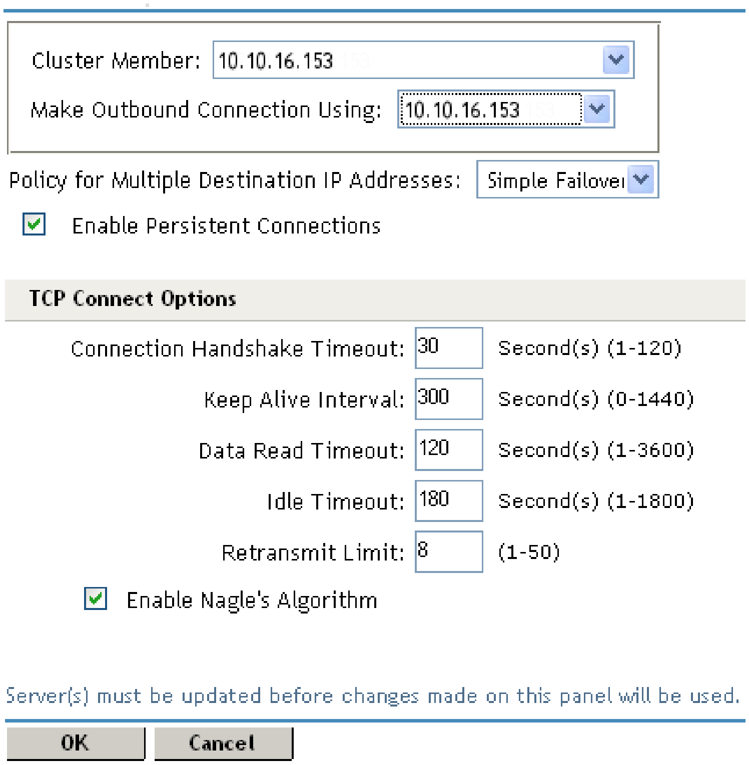
   Enforce 128-Bit Encryption between Access Gateway and Web Server: When this option is selected, the Access Gateway requires all its client connections to Web servers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.

5. To save your changes to browser cache, click *OK*.
6. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.6.2 Configuring TCP Connect Options for Web Servers

Connect options are specific to the group of Web servers configured for a proxy service. They allow you to control how idle and unresponsive Web server connections are handled and to optimize these processes for your network. For most networks, the default values provide adequate performance. If your network is congested and slow, you might want to increase some of the limits.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Web Servers* > *TCP Connect Options*.


Cluster Member: 10.10.16.153

Make Outbound Connection Using: 10.10.16.153

Policy for Multiple Destination IP Addresses: Simple Failove

☑ Enable Persistent Connections

TCP Connect Options

Connection Handshake Timeout: 30 Second(s) (1-120)

Keep Alive Interval: 300 Second(s) (0-1440)

Data Read Timeout: 120 Second(s) (1-3600)

Idle Timeout: 180 Second(s) (1-1800)

Retransmit Limit: 8 (1-50)

☑ Enable Nagle's Algorithm

Server(s) must be updated before changes made on this panel will be used.

2 (Linux only) Configure the IP address to use when establishing connections with Web servers. Configure the following:

Cluster Member: (Available only if the Linux Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. Only the value of the *Make Outbound Connection Using* option applies to the selected server.

Make Outbound Connection Using: (Linux only) Specifies which IP address the proxy service should use when establishing connections with the back-end Web servers.

3 (Linux only) Select how the Web servers should be contacted when multiple Web servers are available. Select one of the following:

- **Simple Failover:** Allows the next available Web server in the group to be contacted when the first server in the list is no longer available.
- **Round Robin:** Moves in order through the list of Web servers, allowing each to service requests before starting at the beginning of the list for a second group of requests.

  This is the default behavior of the NetWare Access Gateway, and it is not configurable.

4 Select *Enable Persistent Connections* to allow the Access Gateway to establish a persistent HTTP connection between the Access Gateway and the Web server. Usually, HTTP connections service only one request and response sequence. A persistent connection allows multiple requests to be serviced before the connection is closed.

This option is enabled by default.

5 To modify the connection timeouts between the Access Gateway and the Web servers, configure the following fields:

Connection Handshake Timeout: Sets a timeout limit for a connecting device that stops responding after initiating the TCP handshake process. If an expected handshake response is not received from the connecting device in this amount of time, an error occurs. Setting the value lower might help defend against SYN attacks. The timeout can be set from 1 to 120 seconds. The default is 30 seconds.

Keep Alive Interval: Sets the length of time between packets being sent to a connected device to determine if the connection is still alive. If a response is not received within the Data Read Timeout value, the connection is closed. On an idle connection, sending these ping packets continues until the Idle Timeout value is reached. Setting the value to zero prevents the sending of keep-alive packets. The value can be set from 0 to 1440 seconds (24 minutes). The default is 300 seconds (5 minutes).

Data Read Timeout: Determines when an unresponsive connection is closed. When exchanging data, if an expected response from the connected device is not received within this amount of time, the connection is closed. This value might need to be increased for slow or congested network links. The value can be set from 1 to 3600 seconds (1 hour). The default is 120 seconds (2 minutes).

Idle Timeout: Determines when an idle connection is closed. If no application data is exchanged over a connection for this amount of time, the connection is closed. This value limits how long an idle persistent connection is kept open. This setting is a compromise between freeing resources to allow additional inbound connections, and keeping connections established so that new connections from the same device do not need to be re-established. The value can be set from 1 to 1800 seconds (30 minutes). The default is 180 seconds (3 minutes).

Retransmit Limit: Determines how many times data is resent. When exchanging data, if the expected acknowledgement (ACK) response is not received, this is the number of times the device attempts to resend the data before closing the connection. You can set the value from 1 to 50. The default is 8.

Enable Nagle's Algorithm: Determines whether small buffer messages can be concatenated into one large message. When this option is enabled, small buffer messages are automatically concatenated. This process increases the efficiency of a network application system by decreasing the number of packets that must be sent. Enabling this feature delays data transmission until a full TCP packet can be sent.

6. To save your changes to browser cache, click *OK*.
7. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

13.6.3 Configuring Connection and Session Persistence

The Access Gateway establishes three types of connections:

- Access Gateway to browser
- Access Gateway to Web server
- Browser to Web server

The Access Gateway to the browser connections and the Access Gateway to the Web server connections involve setting up a TCP connection for an HTTP request. HTTP connections usually service only one request and response sequence, and the TCP connection is opened and closed during the sequence. A persistent connection allows multiple requests to be serviced before the connection is closed and saves a significant amount of processing time. To configure this type of persistence, see the following:

- **Access Gateway to Browser:** Click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *TCP Listen Options* and configure the *Enable Persistent Connections* option.

- **Access Gateway to Web Server:** Click *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > Web Servers > TCP Connect Options* and configure the *Enable Persistent Connections* option.

The persistence of the browser to Web server connection is always enabled and is not configurable. This feature allows a browser to use the same Web server after an initial connection has been established. Most Web applications are designed to expect this type of behavior.

13.6.4 Configuring the Session Timeout

When a user logs in and authenticates to the Identity Server, the Identity Server establishes a session for the user and sets an inactivity timeout for the session. If the user's session becomes inactivity and reaches this time limit, the session becomes invalid. If the user tries to access a resource from an invalid session, the user is prompted to log in again.

The session timeout is a global value, affecting all users who authenticate to the Identity Server and all resources protected by Access Manager. The default value for the session timeout is 15 minutes.

To modify this value:

1. In the Administration Console, click *Access Manager > Identity Servers > Edit*.
2. For the *Session timeout* option, use the up-arrow button to increase the timeout and the down-arrow button to decrease the timeout.
3. Click *OK*, then update the Identity Server.

Configuring the Access Gateway for SSL

14

SSL provides the following security features:

- Authentication and nonrepudiation of the server through the use of digital signatures
- Data confidentiality through the use of encryption
- Data integrity through the use of authentication codes

Mutual SSL provides the same things as SSL, with the addition of authentication and nonrepudiation of the client, by using digital signatures.

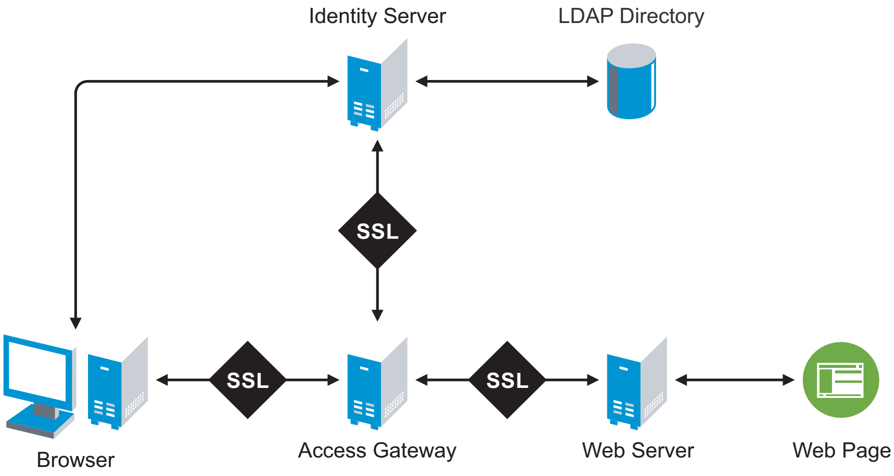
To ensure the validity of X.509 certificates, Access Manager supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) methods of verification.

This section describes how the Access Gateway can use SSL in its interactions with other Access Manager components, how you can enable SSL between an Access Gateway and these components, and how you can use other options to increase security:

14.1 Using SSL on the Access Gateway Communication Channels

You can configure the Access Gateway to use SSL in its connections to the Identity Server, to the browsers, and to its Web servers. Figure 14-1 illustrates these communication channels.

Figure 14-1 Setting Up SSL for the Access Gateway Communication Channels



This section only describes how to set up SSL for the Access Gateway communication channels. The Identity Server needs to be configured for SSL before the Access Gateway can be configured for SSL. See "Configuring Secure Communication on the Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

When the user logs in to the Identity Server, the Identity Server verifies the user's credentials, usually with the credentials stored in an LDAP directory, but other methods are available. If the login is successful, the Identity Server sends an artifact to the browser, and the browser forwards it to the Access Gateway. The Access Gateway uses the artifact to retrieve the user's name and password from the Identity Server. The Access Gateway and Identity Server channel is probably the first communication channel you should enable for SSL. The Access Gateway uses an embedded service provider to communicate with the Identity Server. When you enable SSL between the two, the Access Manager distributes the necessary certificates to set up SSL. However, if you have configured the Identity Server to use certificates from an external certificate authority (CA), you need to import the public certificate of this CA into the trust store of the Access Gateway. If you have set up the Access Gateway to use a certificate from an external CA, you need to import the public certificate of this CA into the trust store of the Identity Server.

SSL must be enabled between the Access Gateway and the browsers before you can enable SSL between the Access Gateway and its Web servers. If you enable SSL between the Access Gateway and the browsers, SSL is automatically enabled for the Access Gateway embedded service provider that communicates with the Identity Server. After you have enabled SSL between the Access Gateway and the browsers, you can select whether to enable SSL between the Access Gateway and the Web servers. By not enabling SSL to the Web servers, you can save processing overhead if the data on the Web servers is not sensitive or if it is already sufficiently protected.

Whether you need the added security of SSL or mutual SSL between the Access Gateway and its Web servers depends upon how you have set up your Web servers. If you have configured the Web servers so that they can only accept connections with the Access Gateway, mutual SSL is probably not needed. On the other hand, if the Access Gateway is injecting authentication credentials into HTTP headers, you should enable SSL.

14.2 Prerequisites for SSL

The following SSL configuration instructions assume that you have already created or imported the certificate that you are going to use for SSL. This certificate must have a subject name (cn) that matches the published DNS name of the proxy service that you are going to use for authentication. You can obtain this certificate one of two ways:

- You can use the Access Manager CA to create this certificate. See Section 24.1.1, "Creating a Locally Signed Certificate," on page 358.
- You can create a certificate signing request (CSR), send it to an external CA, then import the returned certificates into Access Manager. See Section 24.1.2, "Generating a Certificate Signing Request," on page 365 and Section 24.5, "Importing Public Key Certificates (Trusted Roots)," on page 368.

14.2.1 Prerequisite for SSL Communication between the Identity Server and the Access Gateway

If you are going to set up SSL communication between the Identity Server and the Access Gateway for authentication and you have configured the Identity Server to use certificates created by an external CA, you need to import the public certificate of this CA into the trusted root keystore of the Access Gateway.

1. If you haven't already imported the public certificate of this CA into the trusted root store of the Identity Server, do so now. For instructions, see Section 24.5, "Importing Public Key Certificates (Trusted Roots)," on page 368.
2. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Service Provider Certificates* > *Trusted Roots*.
3. In the *Trusted Roots* section, click *Add*.
4. Click the *Select trusted root(s)* icon, select the public certificate of the CA that signed the Identity Server certificates, then click *OK*.
5. Specify an alias, then click *OK* twice.
6. To apply the changes, click *Close*, then on the Access Gateways page, click *Update*.

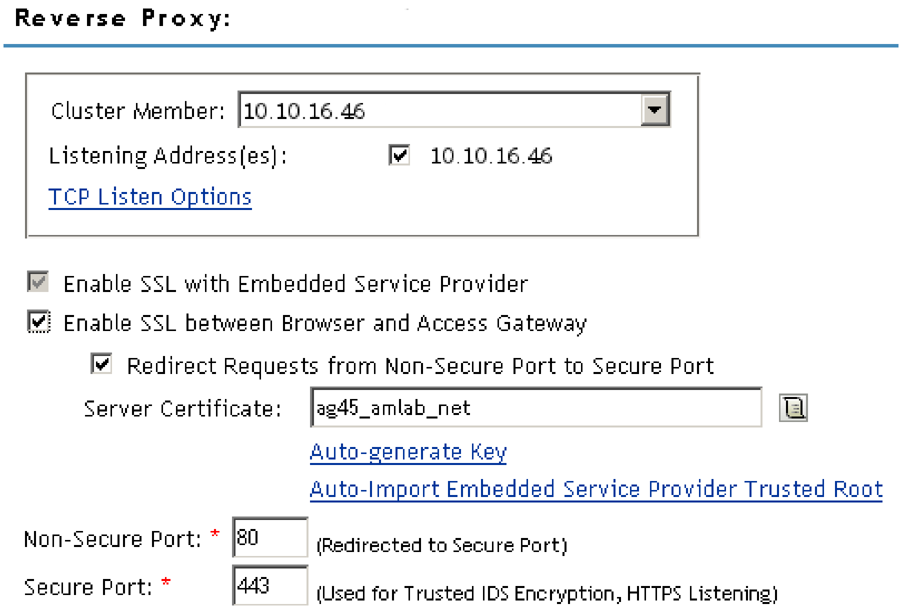
14.2.2 Prerequisites for SSL Communication between the Access Gateway and the Web Servers

If you are going to set up SSL between the Access Gateway and the Web servers, you need to configure your Web servers for SSL. Your Web servers must supply a certificate that clients (in this case, the Access Gateway) can import. See your Web server documentation for information on how to configure the Web server for SSL.

For mutual SSL, the proxy service must supply a certificate that the Web server can trust. This certificate can be the same one you use for SSL between the browsers and the reverse proxy.

14.3 Configuring SSL Communication with the Browsers and the Identity Server

1 In the Administration Console, click *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy]*.



Reverse Proxy:

Cluster Member: 10.10.16.46

Listening Address(es): ☑ 10.10.16.46

☑ Enable SSL with Embedded Service Provider

☑ Enable SSL between Browser and Access Gateway

☑ Redirect Requests from Non-Secure Port to Secure Port

Server Certificate: ag45_amlab_net

Non-Secure Port: * 80 (Redirected to Secure Port)

Secure Port: * 443 (Used for Trusted IDS Encryption, HTTPS Listening)

2 Configure the reverse proxy for SSL. Fill in the following fields:

Enable SSL with Embedded Service Provider: Select this option to encrypt the data exchanged for authentication (the communication channel between the Identity Server and the Access Gateway). This option is only available for the reverse proxy that has been assigned to perform authentication.

If you enable SSL between the browsers and the Access Gateway, this option is automatically selected for you. You can enable SSL with the embedded service provider without enabling SSL between the Access Gateway and the browsers. This allows the authentication and identity information that the Access Gateway and the Identity Server exchange to use a secure channel, but allows the data that the Access Gateways retrieves from the back-end Web servers and sends to users to use a non-secure channel. This saves processing overhead if the data on the Web servers is not sensitive.

Enable SSL between Browser and Access Gateway: Select to require SSL connections between your clients and the Access Gateway. SSL must be configured between the browsers and the Access Gateway before you can configure SSL between the Access Gateway and the Web servers.

Redirect Requests from Non-Secure Port to Secure Port: Determines whether browsers are redirected to the Secure Port and allowed to establish an SSL connection. If this option is not selected, browsers that connect to the non-secure port are denied service.

This option is only available if you have selected *Enable SSL with Embedded Service Provider*.

3. Select the certificate to use for SSL between the Access Gateway and the browsers. Select one of the following methods:
   - To auto-generate a certificate key by using the Access Manager CA, click *Auto-generate Key*, then click *OK* twice. The generated certificate appears in the *Server Certificate* text box.

     The generated certificate uses the published DNS name of the first proxy service for the Subject name of the certificate. If there is more than one proxy service, the CA generates a wildcard certificate (*.Cookie Domain).

     If you have not created a proxy service for this reverse proxy, wait until you have created a proxy service before generating the key. This allows the CN in the *Subject* field of the certificate to match the published DNS name of the proxy service.
   - To select a certificate, click the *Select Certificate* icon, select the certificate you have created for the DNS name of your proxy service, then click *OK*. The certificate appears in the *Server Certificate* text box. For SSL to work, the CN in the Subject field of the certificate must match the published DNS name of the proxy service.
4. (Conditional) If you have selected a certificate in Step 3 that was created by an external CA, click *Auto-Import Embedded Service Provider Trusted Root*, click *OK*, specify an alias name, click *OK*, then click *Close*.

   This option imports the public key from the embedded service provider into the trust store of the Identity Servers in the selected Identity Server Configuration. This sets up a trusted SSL relationship between the Identity Server and the embedded service provider.
5. Configure the ports for SSL:

   Non-Secure Port: Specifies the port on which to listen for HTTP requests. The default port for HTTP is 80.
   - If you have selected the *Redirect Requests from Non-Secure Port to Secure Port* option, requests sent to this port are redirected to the secure port. If the browser can establish an SSL connection, the session continues on the secure port. If the browser cannot establish an SSL connection, the session is terminated.
   - If you do not select the *Redirect Requests from Non-Secure Port to Secure Port* option, this port is not used when SSL is enabled.

   IMPORTANT: If you select not to redirect HTTP requests (port 80) and your Access Gateway has only one IP address, do not use port 80 to configure another reverse proxy. Although it is not used, it is reserved for this reverse proxy.

   Secure Port: Specifies the port on which to listen for HTTPS requests (usually 443). This port needs to match the configuration for SSL. If SSL is enabled, this port is used for all communication with the browsers. The listening address and port combination must not match any combination you have configured for another reverse proxy or tunnel.
6. Click *OK*.
7. On the *Configuration* page, click *Reverse Proxy / Authentication*.
8. In the *Embedded Service Provider* section, click *Auto-Import Identity Server Trusted Root*, click *OK*, specify an alias, click *OK* twice, then click *Close*.

   This option imports the public key of the Identity Server into the trust store of the embedded service provider. This sets up a trusted SSL relationship between the embedded service provider and the Identity Server.

The configCA public key certificate of the Access Manager CA is automatically added to the ESP Trust Store. If you are using Access Manager CA certificates for the Identity Server, you do not need to import the configCA certificate unless someone has deleted it from this trust store.

9 Click *OK*.

10 On the Server Configuration page, click *OK*.

11 On the Access Gateways page, click *Update* > *OK*.

The embedded service provider is restarted during the update.

12 Update the Identity Server so that it uses the new SSL configuration. Click *Identity Servers* > *Update*.

13 Verify that the trusted relationship between the Identity Server and the Access Gateway has been reestablished.

13a Enter the URL to a protected resource on the Access Gateway.

13b Complete one of the following:

- If you are prompted for login credentials, enter them. The trusted relationship has been reestablished.
- If you receive a 100101043 or 100101044 error, the trusted relationship has not been established. For information on solving this problem, see "Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors" in the *Novell Access Manager 3.0 SP3 IR2 Administration Guide*.

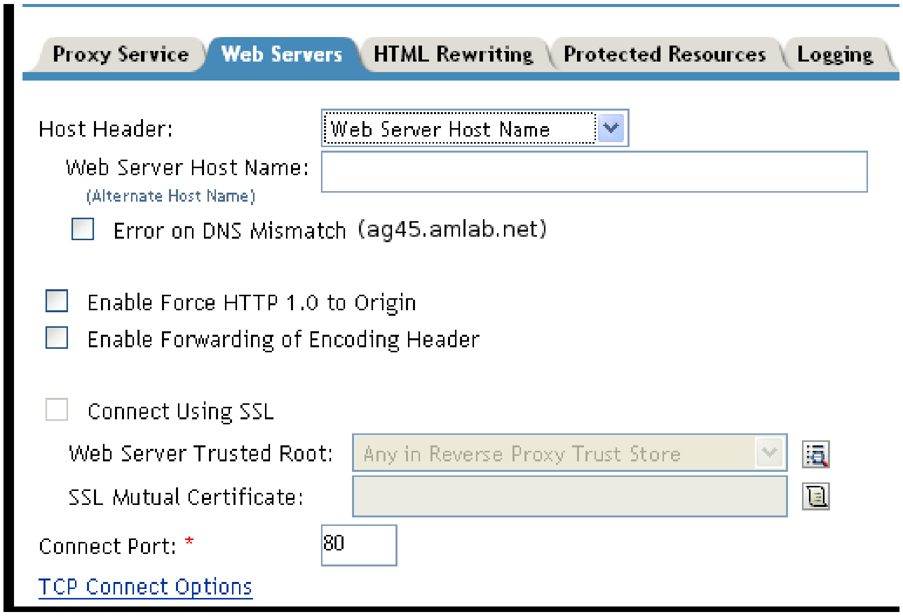
14 To configure additional security options, continue with one of the following:

- Section 14.4, "Configuring SSL between the Proxy Service and the Web Servers," on page 244
- Section 14.6, "Configuring the Encryption Key," on page 248
- Section 14.7, "Avoiding Non-Secure Cookies," on page 248

14.4 Configuring SSL between the Proxy Service and the Web Servers

SSL must be enabled between the Access Gateway and the browsers before you can enable it between the Access Gateway and its Web servers.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Web Servers*.



Host Header: Web Server Host Name

Web Server Host Name:

(Alternate Host Name)

☐ Error on DNS Mismatch (ag45.amlab.net)

☐ Enable Force HTTP 1.0 to Origin

☐ Enable Forwarding of Encoding Header

☐ Connect Using SSL

Web Server Trusted Root: Any in Reverse Proxy Trust Store

SSL Mutual Certificate:

Connect Port: * 80

2 To configure SSL, select *Connect Using SSL*.

This option is not available if you have not set up SSL between the browsers and the Access Gateway. See Section 14.3, "Configuring SSL Communication with the Browsers and the Identity Server," on page 242 and select the *Enable SSL between Browser and Access Gateway* field.

3 In the *Connect Port* field, specify the port that your Web server uses for SSL communication. The following table lists some common servers and their default ports.

Server Type	Non-Secure Port	Secure Port
Web server with HTML content	80	443
SSL VPN	8080	8443
WebSphere	9080	9443
JBoss	8080	8443

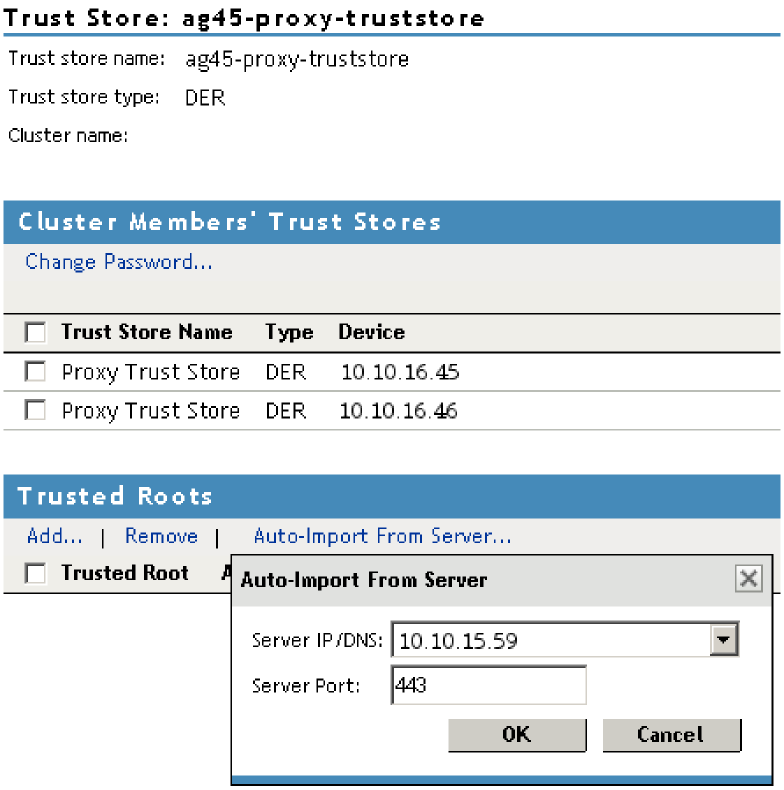
4 Configure how you want the certificate verified. The Access Gateway platforms support different options:

4a (Conditional) If you are configuring a Linux Access Gateway, select one of the following options:

- To not verify this certificate, select *Do not verify* for the *Web Server Trusted Root*. Continue with Step 9.
- To allow the certificate to match any certificate in the trust store, select *Any in Reverse Proxy Trust Store* for the *Web Server Trusted Root*. Continue with Step 9.
- To add a certificate to the trust store for the Web server, click the *Manage Reverse Proxy Trust Store* icon. Continue with Step 4c.

4b (Conditional) If you are configuring a NetWare® Access Gateway, all the certificates in the certificate chain of the Web server must be in its trust store. To add these certificates to the trust store, click *Any in Reverse Proxy Trust Store*. Continue with Step 4c.

4c The auto import screen appears.



Trust Store: ag45-proxy-truststore

Trust store name: ag45-proxy-truststore
Trust store type: DER
Cluster name:

Cluster Members' Trust Stores

Change Password...

☐	Trust Store Name	Type	Device
☐	Proxy Trust Store	DER	10.10.16.45
☐	Proxy Trust Store	DER	10.10.16.46

Trusted Roots

Auto-Import From Server

Server IP/DNS: 10.10.15.59
Server Port: 443

If the Access Gateway is a member of a cluster, the cluster members are listed. The Web server certificate is imported into the trust stores of each cluster member.

5 Ensure that the IP address of the Web server and the port match your Web server configuration.

If these values are wrong, you have entered them incorrectly on the Web server page. Click Cancel and reconfigure them before continuing.

6 Click *OK*.

The server certificate, the Root CA certificate, and any certificate authority (CA) certificates from a chain are listed.

If the whole chain is not displayed, import what is displayed. You then need to manually import the missing parents in the chain. A parent is missing if the chain does not include a certificate where the Subject and the Issuer have the same CN.

7 Specify an alias, then click *OK*.

All the certificates displayed are added to the trust store.

8 Click *Close*.

9 (Optional) For mutual authentication, the Access Gateway platforms support different options:

9a (Conditional) If you are configuring a Linux Access Gateway, you need to select the certificate. Click the *Select Certificate* icon, select the certificate you created for the reverse proxy, then click *OK*.

This is only part of the process. You need to import the trusted root certificate of the CA that signed the proxy service’s certificate to the Web servers assigned to this proxy service. For instructions, see your Web server documentation.

9b (Conditional) If you are configuring a NetWare Access Gateway, the text box displays the certificate that is sent to the Web server if the Web server requires it. If the Web server is not set up for mutual SSL, the certificate is not sent.

To set up the Web server for mutual SSL, you need to import the trusted root certificate of the CA that signed the certificate displayed in the text box. For instructions, see your Web server documentation.

10 To save your changes to browser cache, click *OK*.

11 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

14.5 Managing Access Gateway Certificates

- Section 14.5.1, "Managing Embedded Service Provider Certificates," on page 247
- Section 14.5.2, "Managing Reverse Proxy and Web Server Certificates," on page 248

14.5.1 Managing Embedded Service Provider Certificates

The Access Gateway uses an embedded service provider to communicate with the Identity Server. The Service Provider Certificates page allows you to view the private keys, certificate authority (CA) certificates, and certificate containers associated with this module. These keystores do not contain the certificates that the Access Gateway uses for SSL connections to browsers or to back-end Web servers.

To view or modify these certificates:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Service Provider Certificates*.

2 Configure the following:

Signing: The signing certificate keystore. Click this link to access the keystore and replace the signing certificate as necessary. The signing certificate is used to sign the assertion or specific parts of the assertion.

Trusted Roots: The trusted root certificate container for the CA certificates associated with the Access Gateway. Click this link to access the trust store, where you can change the password or add trusted roots to the container.

The embedded service provider must trust the certificate of the Identity Server that the Access Gateway has been configured to trust. The public certificate of the CA that generated the Identity Server certificate must be in this trust store. If you configured the Identity Server to use a certificate generated by a CA other than the Access Manager CA, you must add the public certificate of this CA to the Trusted Roots store. To import this certificate, click *Trusted Roots*, then in the *Trusted Roots* section, click *Auto-Import From Server*. Fill in the IP address or DNS name of your Identity Server and its port, then click *OK*.

You can also auto import the Identity Server certificate by select the *Auto-Import Identity Server Configuration Trusted Root* option on the *Reverse Proxies / Authentication* page (click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxies / Authentication*). With this option, you do not need to specify the IP address and port of the Identity Server.

3 To save your changes to browser cache, click *OK*.

4 To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

14.5.2 Managing Reverse Proxy and Web Server Certificates

You select Access Gateway certificates on two pages in the Administration Console:

- *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy]*
- *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > Web Servers*

When configuring certificates on these pages, you need to be aware that there are two phases that are used to push the certificates into active use.

Phase 1: When you select a certificate on one of these pages, then click *OK*, the certificate is placed in the keystore on the Administration Console and it is pushed to the Access Gateway. The certificate is available for use, but it is not used until you update the Access Gateway.

Phase 2: When you select to update the Access Gateway, the configuration for the Access Gateway is modified to contain references to the new certificate and the configuration change is sent to the Access Gateway. The Access Gateway loads and uses the new certificate.

14.6 Configuring the Encryption Key

You can specify the size of the encryption key that the Access Gateway requires when it establishes connections. For the Linux Access Gateway, these options are set per reverse proxy (see Section 13.6.1, "Configuring TCP Listen Options for Clients," on page 234). For the NetWare Access Gateway, these options are set globally for all reverse proxies.

1. In the Administration Console, click *Access Manager > Access Gateways > Edit > Security Options*.
2. Configure the following fields:

   Enforce 128-Bit Encryption between Browser and Access Gateway: When this option is selected, the Access Gateway requires all its server connections with client browsers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.

   Enforce 128-Bit Encryption between Access Gateway and Web Server: When this option is selected, the Access Gateway requires all its client connections to Web servers to use 128-bit encryption. If the encryption key is less than 128, regardless of the cipher suite, the connection is denied.
3. To save your changes to browser cache, click *OK*.
4. To apply your changes, click the *Access Gateways* link, then click *Update > OK*.
5. To have the Access Gateway apply the changes, you must restart the Access Gateway. On the Access Gateways page, select the server, click *Reboot > OK*.

14.7 Avoiding Non-Secure Cookies

An attacker can spoof a non-secure browser into sending a JSESSION cookie that contains a valid user session. This might happen because the Access Gateway communicates with its embedded service provider on port 8080, which is a non-secure connection. Because the embedded service provider does not know whether the Access Gateway is using SSL to communicate with the browsers, the embedded service provider does not mark the JSESSION cookie as secure when it creates the cookie. The Access Gateway receives the Set-Cookie header from the embedded service provider and passes it back to the browser. Now, there is a non-secure, clear-text cookie sitting in the browser. If an attacker spoofs the domain of the Access Gateway, the browser sends the non-secure JSESSION cookie over a non-secure channel where the cookie might be sniffed.

To stop this from happening, you must first configure Access Gateway to use SSL. See Section 14.3, "Configuring SSL Communication with the Browsers and the Identity Server," on page 242. After you have SSL configured, you need to configure Tomcat to secure the cookie:

1. On the Linux Access Gateway machine, log in as `root`.
2. Change to the `/var/opt/novell/tomcat4/conf` directory.
3. In a text editor, open the `server.xml` file.
4. Search for the connector on port 8080.
5. Add the following parameter to this connector:

   ```
   secure="true"
   ```

   These lines should look similar to the following:

   ```
   <Connector port="8080"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" acceptCount="100"
      debug="0" connectionTimeout="20000"
      disableUploadTimeout="true" secure="true" />
   ```

6. Save the `server.xml` file.
7. Restart Tomcat by entering the following command:

   ```
   /etc/init.d/novell-tomcat4 restart
   ```

Server Configuration Settings

15

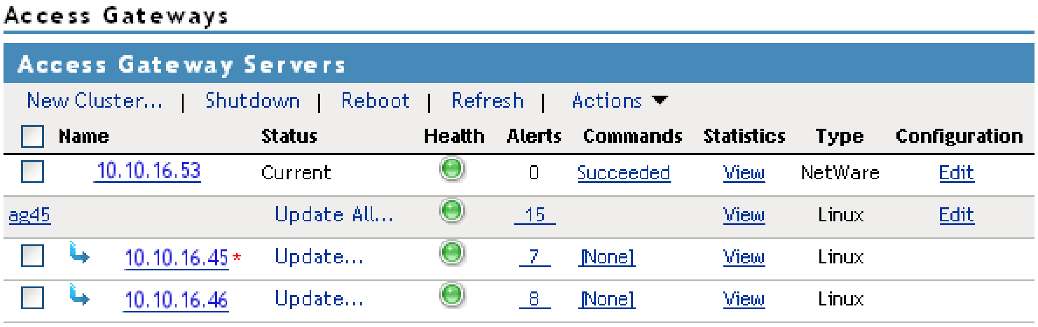
This section describes the configuration settings that affect the Access Gateway as a server, such as changing its name or setting the time.

- Section 15.1, “Viewing and Updating the Configuration Status,” on page 251
- Section 15.2, “Changing the Name of an Access Gateway and Modifying Other Descriptive Details,” on page 252
- Section 15.3, “Setting Date and Time,” on page 253
- Section 15.4, “Setting Up a Tunnel,” on page 255
- Section 15.5, “Customizing Error Pages,” on page 256
- Section 15.6, “Configuring Console Access,” on page 261
- Section 15.7, “Configuring Network Settings,” on page 263
- Section 15.8, “Customizing Log Out,” on page 272
- Section 15.9, “Configuring X-Forwarded-For Headers,” on page 273
- Section 15.10, “Upgrading the Access Gateway Software,” on page 273
- Section 15.11, “Exporting and Importing an Access Gateway Configuration,” on page 274

For logging and audit options, see Section 32.4, “Configuring Access Gateway Logging,” on page 523 and Section 31.3, “Enabling Access Gateway Audit Events,” on page 510.

15.1 Viewing and Updating the Configuration Status

1 In the Administration Console, click Access Manager > Access Gateways.



☐ Name	Status	Health	Alerts	Commands	Statistics	Type	Configuration
☐ 10.10.16.53	Current	●	0	Succeeded	View	NetWare	Edit
ag45	Update All...	●	15		View	Linux	Edit
☐ ↳ 10.10.16.45*	Update...	●	7	[None]	View	Linux	
☐ ↳ 10.10.16.46	Update...	●	8	[None]	View	Linux	

2 View the *Status* column.

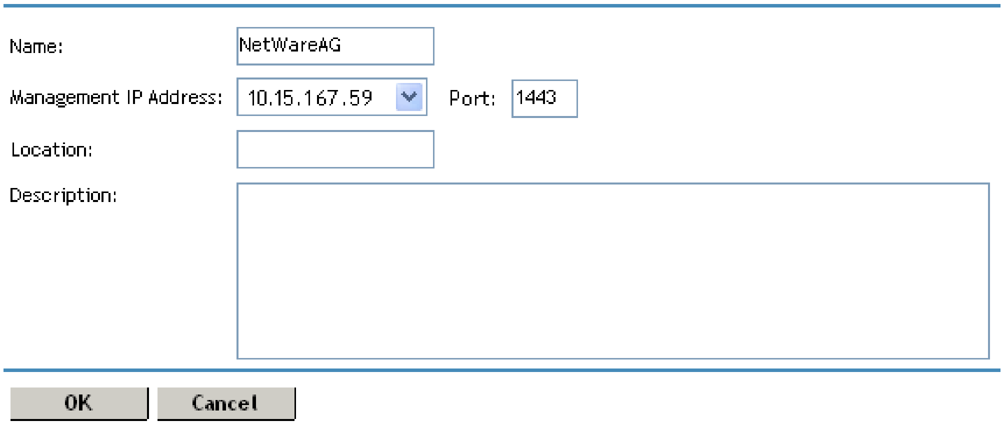
Status	Description
Current	Indicates that all configuration changes have been applied.

Status	Description
Update	Indicates that a configuration change has been made, but not applied. Click this link to apply the changes. ◆ **All Configuration:** You can select to have the server read its complete configuration file. Depending upon what has been modified, updating the complete configuration might cause logged in users to lose data and their connection. ◆ **Logging Settings:** When the ESP logging settings have been modified on the Identity Server, the update option for *Logging Settings* is available. The *Logging Settings* option causes no interruption in services. When you modify Access Gateway logging settings, this option is not available because they are considered configuration settings. ◆ **Policy Settings:** If a policy is modified that the server has enabled for a protected resource and the policy change is the only modification that has occurred, the update option for *Policy Settings* is available. This option causes no interruption in services.
Update All	Available when a server belongs to a cluster. You can select to update all the servers at the same time with the changes, or you can select to update them one at a time. If the modification is a policy or a logging change, then use Update All. If the modification is a configuration change that might interrupt service, we recommend that you update the servers one at a time. For more information, see Section 17.4.3, "Applying Changes to Cluster Members," on page 308.
Pending	Indicates that the server is processing a configuration change, but has not completed the process.

15.2 Changing the Name of an Access Gateway and Modifying Other Descriptive Details

The default name of an Access Gateway is its IP address. You can change this to a more descriptive name as well as adding other details that can help you identity one Access Gateway from another.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Access Gateway]* > *Edit*.



Name: NetWareAG
Management IP Address: 10.15.167.59 Port: 1443
Location:
Description:

2 Modify the values in the following fields:

Name: Specifies the Administration Console display name for the Access Gateway. This is a required field. The default name is the IP address of the Access Gateway. If you modify the name, the name must use alphanumeric characters and can include spaces, hyphens, and underscores.

Management IP Address: Specifies the IP address used to manage the Access Gateway. Select one from the list. For information on changing the *Management IP Address*, see Section 4.3, "Changing the IP Address of the Access Gateway," on page 46.

Location: Specifies the location of the Access Gateway server. This is optional, but useful if your network has multiple Access Gateway servers.

Description: Describes the purpose of this Access Gateway. This is optional, but useful if your network has multiple Access Gateways.

3 Click *OK* twice, then click *Close*.

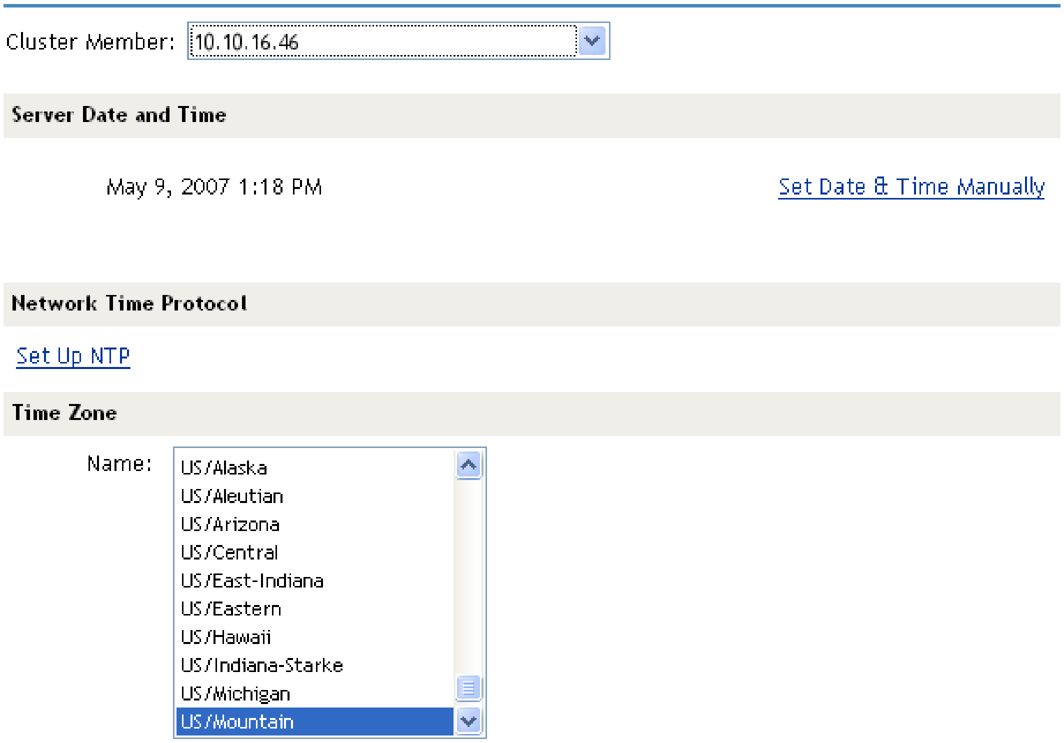
When you click *OK*, any changes are immediately applied to the Access Gateway.

15.3 Setting Date and Time

The *Date & Time* option lets you set the system time for the Access Gateway. The time between the Identity Server and the Access Gateway must be either synchronized or set to be within 1 minute of each other for trusted authentication to work.

To configure the date and time options:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Date & Time*.



Cluster Member: 10.10.16.46

Server Date and Time

May 9, 2007 1:18 PM

2 (Conditional) If the Access Gateway belongs to a cluster of Access Gateways, select the Access Gateway from the list displayed in the *Cluster Member* field. The modifications you make on this page apply only to the selected Access Gateway.

If the Access Gateway does not belong to a cluster, this option is not available.

3 Fill in the following fields:

Server Date and Time: Displays the current time and allows you to set the current time. Click *Set Date & Time Manually*, then select the current year, month, day, hour, and minute.

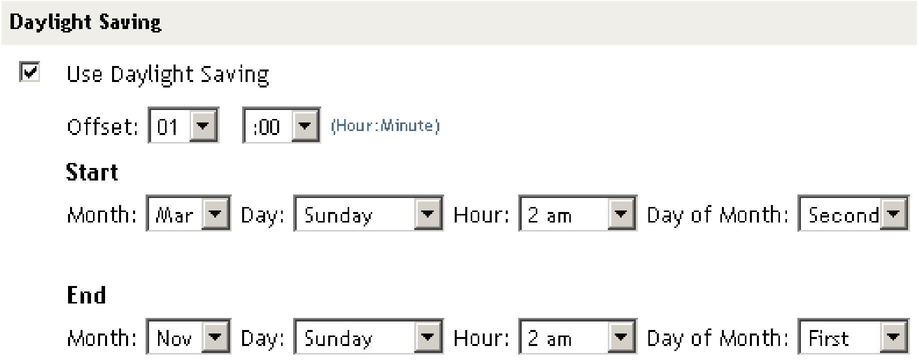
WARNING: If the date is set to a time before the Access Gateway certificates are valid, communication to the Access Gateway is lost. This error cannot be corrected from the Administration Console. You need to correct it at the console of the Access Gateway machine.

- For a NetWare® Access Gateway server, see Section 40.3.3, "Setting the Date and Time at the Console," on page 658.
- For the Linux Access Gateway, use the `yast` command and select *System* > *Date and Time*.

Set Up NTP: Click this option to specify the DNS name or IP address of a Network Time Protocol server. The installation program enters the name of pool.ntp.org, the DNS name of a public NTP server. To disable this feature, you must remove all servers from the NTP Server List. This is not recommended.

Time Zone: Select your time zone, then click *OK*. Regardless of the method you used to set the time, you must select a time zone.

4 (NetWare only) Configure daylight saving time.



Daylight Saving

☑ Use Daylight Saving

Offset: 01 :00 (Hour:Minute)

Start

Month: Mar Day: Sunday Hour: 2 am Day of Month: Second

End

Month: Nov Day: Sunday Hour: 2 am Day of Month: First

Configure the following options:

Use Daylight Saving: Select this option to enable daylight saving.

Offset: Select the hours and minutes that daylight saving varies from standard time.

Start: Select the month, day, hour, and day of month when daylight saving starts.

End: Select the month, day, hour, and day of month when daylight saving ends.

5 To save your changes to browser cache, click *OK*.

6 On the Server Configuration page, click *OK*.

7 To apply your changes, click *Update* > *OK*.

8 (Conditional) If you changed a NetWare Access Gateway from a time zone that uses daylight saving to a time zone that does not use daylight saving, you must reboot the Access Gateway. On the Access Gateway Servers page, select the server, then click *Reboot*.

15.4 Setting Up a Tunnel

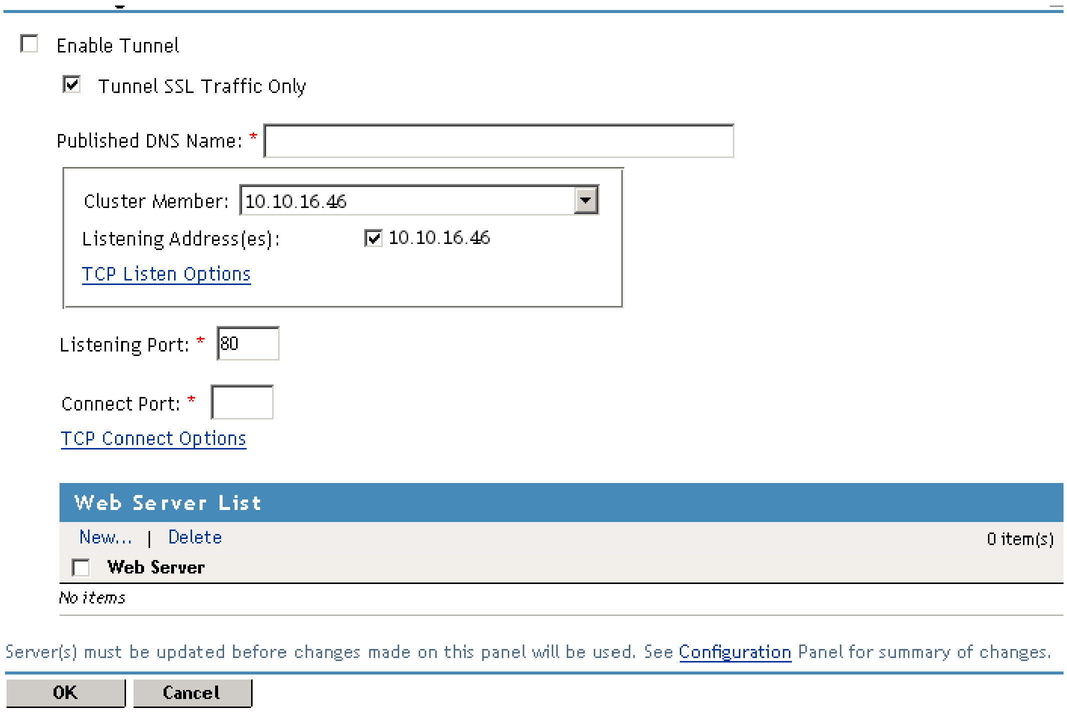
The tunnel option lets you create one or more services for the specific purpose of tunneling non-HTTP traffic through the Access Gateway to the Web server. To do this, the non-HTTP traffic must use a different IP address and port combination than the HTTP traffic.

An Access Gateway usually processes HTTP requests in order to fill them. However, it is not unusual that some of the traffic coming through the gateway is not HTTP-based. Web servers sometimes handle Telnet, FTP, chat, or other kinds of traffic without attempting to process it. If your Web servers are handling this type of traffic, you should set up a tunnel for it.

Reverse proxies and tunnels cannot share the same IP address and port combination. You can either configure a reverse proxy for an IP address and port or a tunnel for that IP address and port.

To set up a tunnel:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Tunneling*.
2. Click *New*, enter a display name for the tunnel, then click *OK*.



☐ Enable Tunnel

☑ Tunnel SSL Traffic Only

Published DNS Name: *

Cluster Member: 10.10.16.46

Listening Address(es): ☑ 10.10.16.46

Listening Port: * 80

Connect Port: *

No items

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

3. Fill in the following fields:

   Enable Tunnel: Specifies that the Access Gateway should set up a tunnel for all incoming traffic. This option must be enabled to configure a tunnel.

   Tunnel SSL Traffic Only: Allows you to configure the Access Gateway to tunnel only SSL traffic. If this option is selected, the Access Gateway verifies that the address and port being accessed are actually an SSL Web site. If verification fails, the service tears down the connection. The SSL port number for the SSL tunnel is specified via the *Listening Port* and the *Connect Port*.

   Published DNS Name: Specify the DNS name you want the public to use to access your tunnel or the virtual IP address assigned to the Access Gateway cluster by the L4 switch. If you specify a DNS name, the DNS name must resolve to the IP address you set up as the listening address for the tunnel.

4. Configure the communication options between the browsers and the tunnel by configuring the following fields:

   Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. The *Listening Address(es)* modifications apply to the selected server. Any other modifications apply to all servers in the cluster.

   Listening Address(es): Displays a list of available IP addresses. If the Access Gateway has only one IP address, only one is displayed. If it has multiple addresses, you can select one or more addresses to enable. You must enable at least one address by selecting its check box.

   TCP Listen Options: Provides additional options for configuring how requests are handled. See Section 13.6.1, “Configuring TCP Listen Options for Clients,” on page 234. At least one Web server must be configured before you can modify these options.

   Listening Port: Specifies the port on which to listen for requests from browsers. The listening address and port combination must not match any combination you have configured for a reverse proxy.

5. Configure the communication options between the tunnel and the Web servers by configuring the following fields:

   Connect Port: Specifies the port that the Access Gateway uses to communicate with the Web server.

   TCP Connect Options: Allows you to control how idle and unresponsive Web server connections are handled and to optimize these processes for your network. See Section 13.6.2, “Configuring TCP Connect Options for Web Servers,” on page 235.

6. Specify a Web server to receive the traffic. In the Web Server List section, click *New*, specify the IP address or DNS name of the Web server, then click *OK*.

   At least one Web server must be specified in the list before you can save a tunnel configuration.

7. To save your changes to browser cache, click *OK*.

8. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

15.5 Customizing Error Pages

- Section 15.5.1, “Customizing Error Pages for the Linux Access Gateway,” on page 256
- Section 15.5.2, “Configuring Error Page Presentation for the NetWare Access Gateway,” on page 261

15.5.1 Customizing Error Pages for the Linux Access Gateway

With the Novell Access Manager 3.0 SP3 release, the Linux Access Gateway uses the custom error page template to rebrand and localize the language of error pages that are published to the browser.

WARNING: With the SP3 release, `error.jsp` files are no longer used to serve error pages. If you have customized the error pages during the previous release by using the `error.jsp` file, you will lose all of your changes. You must now customize your error pages by using the error pages template. The service-based customizing of error pages is not supported with these SP 3 changes.

By default, the Linux Access Gateway contains the following files to help customize and localize the error messages:

- The error page configuration file, `ErrorPagesConfig.xml`
- The error page template file, `ErrorPageTemplate.htm.en`
- The error messages file, `ErrorMessages.xml.en`

NOTE: If you are modifying any of the above files, ensure that you retain the original file names.

The Linux Access Gateway maintains three directories to save files that are used for error page configuration:

```
/var/novell/errorpagesconfig/.factory
/var/novell/errorpagesconfig/.backup
/var/novell/errorpagesconfig/current
```

During the initial installation, the default template files packaged in the build are copied to the `.factory` and the `current` directories. If you have not customized the files in the `current` directory, subsequent installations do not overwrite these files.

When the next version of the files is installed, the files in the `current` directory are copied to the `.backup` directory with the format `<filename>.oldBuildNo`. This ensures that the old build files and customized files are always available in the `.backup` directory.

You can customize and localize the error template and the error messages:

- "Customizing the Error Pages by Using the Default Template" on page 257
- "Customizing and Localizing Error Messages" on page 259

Customizing the Error Pages by Using the Default Template

To customize the default error page template, you must edit the `ErrorPageTemplate.htm.en` file as follows:

NOTE: Make sure that you save the `ErrorPageTemplate.htm.en` file as a backup, before modifying it.

---

1. Log in to a Linux Access Gateway machine.
2. Open the `ErrorPageTemplate.htm.en` file located in the `/var/novell/errorpagesconfig/current` directory.

   A sample error page template is as follows:

```
<html>
     <head><title>Information Alert</title></head>
         <body bgcolor="white"><div align="center"><center>
<table border="0" cellpadding="2" frame height="199"
style="margin-top: 1px; margin-bottom: 1px; padding-top: 1px;
padding-bottom: -1px">
               <tr>
                <td height="34" align="center"><font
color="black" face="Arial Bold" size="4">
                 <b><p align="center"></b></font><font
```

```
face="Intrepid" size="6"               color="#000080">
<strong>Information Alert </strong></font>
                 </td>
               </tr>
               <tr>
                <td height="20" align="center"><img height="8"
width="445" src="<PROXY_ADDRESS>/images/
Odyssey_LoginHead.gif"></td>
              </tr>
              <tr>
    <td height="24" width="444" bgcolor="white" align="center">
    <p align="left">
         <b><br><font color="black" face="Comic Sans
MS">Status</font></b>
            <font color="#ff0033" face="Comic Sans MS"><b>: </
b></font><font color="black" face="Comic Sans
MS"><ERROR_STATUS> </font>
                    </p>
              <p align="left">
                <font color="black" face="Comic Sans
MS"><b>Description</b></font>
                       <font color="#ff0033" face="Comic Sans
MS"><b>: </b></font>
      <font color="black" face="Comic Sans
MS"><ERROR_DESCRIPTION></font>
            </p>    <br>    <br>
     </font></td>
      </tr>
     <tr><td width="444" height="10" align="center"><img
height="8" width="445" src="<PROXY_ADDRESS>/images/
LAP_interoperable_logo_100.gif"></td></tr>
</table>
</center></div>
</body>
</html>
```

3. Modify the error page template. You can edit the default template to customize the user interface, embedded images and to provide localization. However, <ERROR_STATUS> and ERROR_DESCRIPTION> tags should not be removed because, the following actions take place when the error page is served to the browser:
   - `<ERROR_STATUS>`: When the When the error page is served to the browser, `<ERROR_STATUS>` is replaced with the HTTP status code description.
   - `<ERROR_DESCRIPTION>`: When the error page is served to the browser, `<ERROR_DESCRIPTION>` is replaced with the detailed error description.

   If you have changed the images with a new image:
   - All the images must be linked to the *`<PROXY_ADDRESS>`*`/images/` directory.
   - All the images must be copied to Tomcat in the path `/var/opt/novell/tomcat4/webapps/LAGERROR/images`.

   If you have changed an image but retained the filename, press Ctrl + F5 in the browser to refresh the Linux Access Gateway cache.

4 Save the file.

5 Enter the following commands to restart the machine:

```
/etc/init.d/novell-vmc stop
/etc/init.d/novell-vmc start
```

Customizing and Localizing Error Messages

When the Linux Access Gateway serves an error message to the browser by using the `Accept-Language` header value received from the browser, it selects a suitable error template and an error message file. To localize the error messages, you must to do the following:

- Localize or customize the error messages in the `ErrorPagesConfig.xml` file and save it with the language extension. For more information, see "Localizing and Customizing the Error Messages" on page 259.
- Modify the `ErrorPagesConfig.xml` file with the header value and the template mapping information. For more information, see "Modifying the ErrorPagesConfig.XML file" on page 260.

Localizing and Customizing the Error Messages

The error messages contained in the `ErrorMessages.xml.en` file can be localized in various languages and stored as `ErrorMessages.xml.`*<lang>*, where *<lang>* is the `fileXn` attribute value. You can also customize the English error messages present in the `ErrorMessages.xml.en` file.

NOTE: You cannot customize an error message that is not present in the `ErrorMessages.xml.en` file.

To localize the error messages:

1 Log in as `root`.

2 Open the `ErrorMessages.xml.<lang>` file.

3 Copy the error messages that you have localized or customized to within the `<TranslatedMessage></TranslatedMessage>` tags. For example:

```
</Message>
      <Message id="<ID No>" name="<ERROR_MESSAGE_NAME>"
enable="yes">
            <EnglishMessage>English Message goes here</
EnglishMessage>
<TranslatedMessage>
Localized message goes here
</TranslatedMessage>
</Message>
```

NOTE: Do not delete the contents within the `<TranslatedMessage></TranslatedMessage>` tags from an English file because, the `ErrorPagesConfig.xml` file selects the error message within these tags for display.

4 Enter the following commands to restart the Linux Access Gateway:

```
/etc/init.d/novell-vmc stop
/etc/init.d/novell-vmc start
```

Modifying the ErrorPagesConfig.XML file

The `ErrorPagesConfig.xml` file stores the header value and the template mapping information. You must edit the `ErrorPagesConfig.xml` file to provide localization for error messages in various languages. In the `ErrorPagesConfig.xml` file, each `<Profile>` element corresponds to a template file `ErrorPageTemplate.htm.<lang>` and a messages file `ErrorMessages.xml.`*`<lang>`*, where *<lang>* is the `fileXn` attribute value. For example, if the `fileXn` attribute value is `de`, the `ErrorPagesTemplate.htm.de` file is served to the browser.

To map a list of `Accept-Language` header values to the template, you must add the header value as the `<header>` element under the corresponding `<Profile>` element.

To modify the `ErrorPagesConfig.xml` file:

1 Log in to a Linux Access Gateway machine.

2 Open the `ErrorPagesConfig.xml` file located in the `/var/novell` directory.

3 Add the language information within the `<profile>` tag as follows:

```
<ErrorPageConfiguration>
      <Profile name = “English”  enable = “1”  fileXn = “en”>
               <header value = “en-us” />
                <header value = “en-uk” />

               <header value = “en-any” />
               <header value = “any” />
       </Profile>
       <Profile name = “German” enable = “1” fileXn = “de”>
               <header value = “de-CH” />
               <header value = “de-any” />
       </Profile>
</ErrorPageConfiguration>
```

This file serves the error messages from:

- The English profile, if the header value is en-us or en-uk or en-*
- The German profile, if the header value is de-CH or de-*
- The default profile, if the header value is not any of the above, or if it is defined as `any`.

When the header value is defined as `any`, the default profile is served. This profile matches any header value that did not have a matching profile. For example, if the header value entry is `en-any`, and the `Accept-Language` header value of the browser is `en-xyz` (for which there is no proper match), then the profile with the entry `en-any` would be a match.

If `any` is used to search for any language-specific files, then the word `any` must be preceded by the hyphen (-). For example, you must not specify `en-cany` as the header value entry to match en-c* header values.

4 Save the file.

5 Enter the following commands to restart the machine:

```
/etc/init.d/novell-vmc stop
/etc/init.d/novell-vmc start
```

15.5.2 Configuring Error Page Presentation for the NetWare Access Gateway

The Error Page option allows you to specify how the error pages generated by the NetWare Access Gateway are published to the browsers.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Error Page*.
2. In the *Error Page Language* field, select the language in which the error page is published.
3. To save your changes to browser cache, click *OK*.
4. On the Server Configuration page, click *Close*, then click *Update* > *OK*.

WARNING: The messages are available from Novell in English. To use another language, you must translate the messages to that language. For information on how to customize the error messages for a specific language, see "Customizing Error Pages" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

15.6 Configuring Console Access

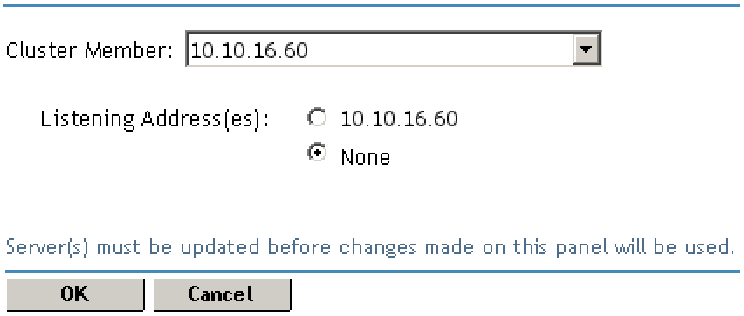
The following options control access to the NetWare Access Gateway console:

- Section 15.6.1, "Setting Up an FTP Listening Address," on page 261
- Section 15.6.2, "Enabling Console Access with SSH and Telnet Sessions," on page 262
- Section 15.6.3, "Setting the Password for the admin and config Users," on page 263

15.6.1 Setting Up an FTP Listening Address

(NetWare only) The Mini FTP option allows you to configure an FTP listening address for management. If this option is enabled, you can use FTP to upload and download files.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Mini FTP*.



Cluster Member: 10.10.16.60

Listening Address(es): ○ 10.10.16.60 ● None

Server(s) must be updated before changes made on this panel will be used.

2. Fill in the following fields:

   Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. All changes made to this page apply to the selected server.

   Listening Addresses: To enable this feature, select an IP address for FTP listening.

If the Access Gateway server has only one IP address, only one is displayed for selection. If the server has multiple IP addresses, you can select one or more.

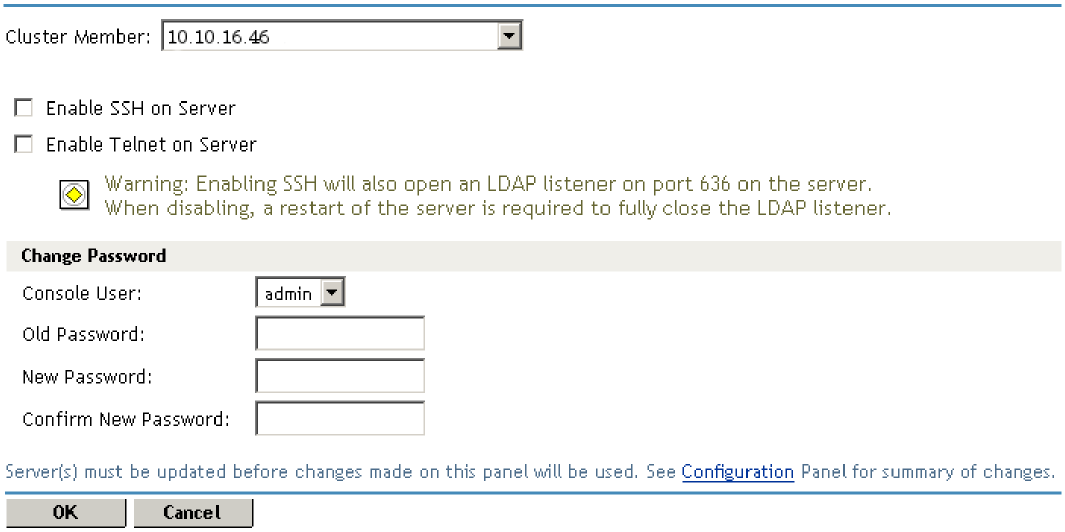
3. To save your changes to browser cache, click *OK*.
4. On the Server Configuration page, click *OK*, then click *Update* > *OK*.

When logging in to an FTP session, the username must be `config`, and the password is empty unless you have configured a password. If you enable FTP, we strongly recommend that you set up a password for the `config` user. See Section 15.6.3, "Setting the Password for the admin and config Users," on page 263.

15.6.2 Enabling Console Access with SSH and Telnet Sessions

(NetWare only) The Console Access option allows you to control whether administrators can set up SSH or Telnet sessions with the NetWare Access Gateway and use command line options to configure it.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Console Access*.



Cluster Member: 10.10.16.46

☐ Enable SSH on Server

☐ Enable Telnet on Server

Warning: Enabling SSH will also open an LDAP listener on port 636 on the server. When disabling, a restart of the server is required to fully close the LDAP listener.

Change Password

Console User: admin

Old Password:

New Password:

Confirm New Password:

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2. Fill in the following fields:

   Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. All changes made to this page apply to the selected server.

   Enable SSH on Server: If this option is selected, SSH is enabled. SSH sets up a secure, encrypted connection between the Access Gateway and the client. Enabling this option opens an LDAP listener on the Access Gateway for port 636. Disabling this option does not fully close the listener. You must restart the Access Gateway to fully close the LDAP listener.

   Enable Telnet on Server: If this option is selected, Telnet is enabled.

   IMPORTANT: Telnet is inherently insecure. All information is sent in clear text, including passwords.

3. To save your changes to browser cache, click *OK*.

4 On the Server Configuration page, click *OK*, then click *Update* > *OK*.

You can use SSH client software or a terminal window to set up a session. When prompted, log in to the NetWare Access Gateway as the `admin` user.

If you enable Telnet, use the client software on your workstation to set up a session. When prompted, you can log in to the NetWare Access Gateway as either the `config` or the `admin` user.

15.6.3 Setting the Password for the admin and config Users

(NetWare only) Access Manager sets up an admin user when you install the Administration Console, and you are prompted to supply a name for this user. During installation, the NetWare Access Gateway sets up an `admin` and a `config` user, for managing the NetWare Access Gateway console. These names are not configurable.

The `admin` user is the NetWare Access Gateway user that has been created for accessing the machine over SSH. It is assigned a default password of `novell`.

The `config` user is the NetWare Access Gateway user that has been created for accessing the machine over FTP and Telnet. If you enable FTP or Telnet, you should set up a password for the `config` user. When an Access Gateway is installed, the `config` user is not assigned a password.

To set or modify the password for the `config` or `admin` user:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Console Access*.
2. (Conditional) If the Access Gateway is a member of a group, select the server you want to configure from the list of servers. All changes made to this page apply to the selected server.
3. In the *Change Password* section, select the *Console User*, either *config* or *admin*.
4. Fill in the following fields:

   Old Password: Specifies the current password for the console user. When used in conjunction with the *New Password* and *Confirm New Password* fields, this field allows you to change the console password. When the `admin` user was created, it was assigned a default password of `novell`. When you install the NetWare Access Gateway, no password is assigned to the `config` user. To create a password the first time for the `config` user, leave this field blank.

   New Password: Specifies a new password. The password must be at least six characters long.

   Confirm New Password: Retype the new password.
5. To save your changes and have them applied, click *OK*.

   As soon as you click *OK*, the change is sent to the NetWare Access Gateway and the password change is applied. The password is not saved to browser cache, and you do not need to update the Access Gateway configuration.

15.7 Configuring Network Settings

After initial setup, you seldom need to change the network settings unless something in your network changes, such as you add a new gateway or DNS server. This section describes the following tasks:

- Section 15.7.1, “Viewing and Modifying Adapter Settings,” on page 264
- Section 15.7.2, “Viewing and Modifying Gateway Settings,” on page 266

- Section 15.7.3, “Viewing and Modifying DNS Settings,” on page 268
- Section 15.7.4, “Configuring Hosts,” on page 270
- Section 15.7.5, “Adding New Network Interfaces to the Linux Access Gateway,” on page 272

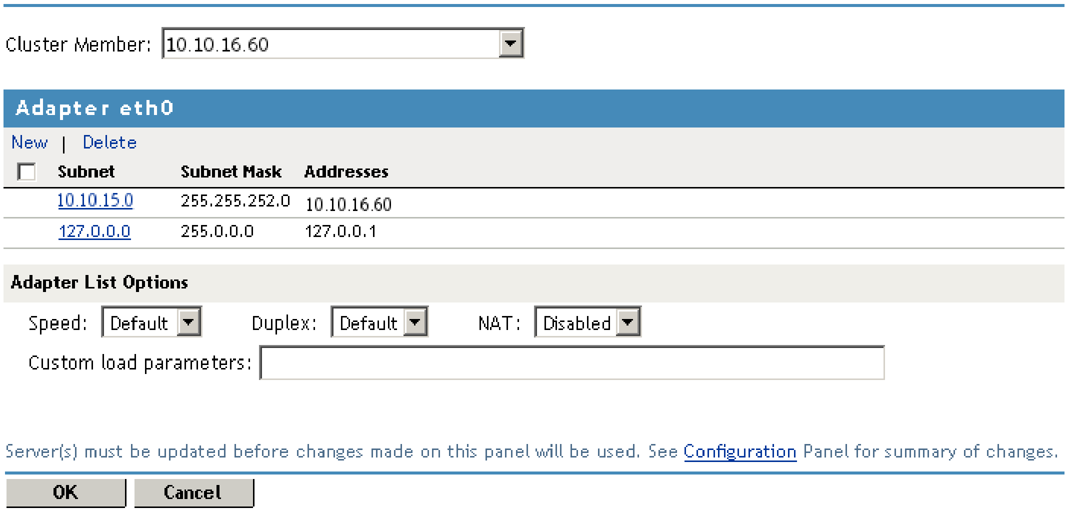
15.7.1 Viewing and Modifying Adapter Settings

The adapter settings allow you to view the current configuration for the network adapters installed in the Access Gateway machine and manage the IP addresses that are assigned to them. If you want to configure an adapter to use more than one IP address, you can use this option to add them.

If you have multiple adapters installed on a Linux Access Gateway machine, you can only configure eth0 during installation. Use the procedure described in this section to configure the others.

To view or modify your current adapter settings:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Adapter List*.



Cluster Member: 10.10.16.60

Adapter eth0

New | Delete

☐	Subnet	Subnet Mask	Addresses
	10.10.15.0	255.255.252.0	10.10.16.60
	127.0.0.0	255.0.0.0	127.0.0.1

Adapter List Options

Speed: Default

Duplex: Default

NAT: Disabled

Custom load parameters:

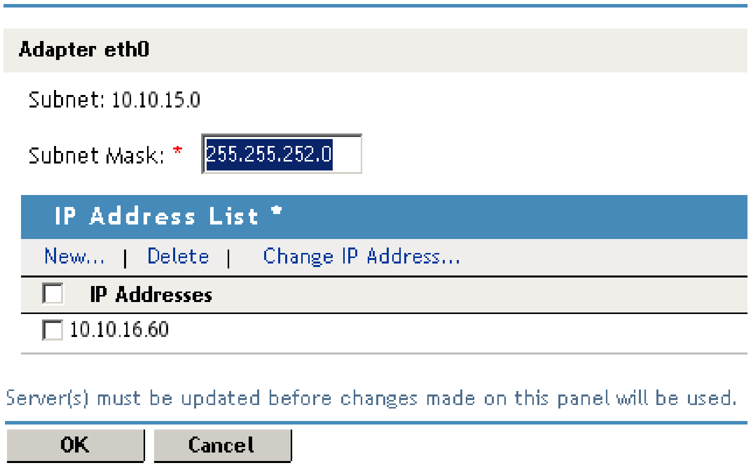
Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the *Cluster Member* field. All changes made to this page apply to the selected server.

3 Select the adapter you want to modify, then select one of the following actions:

- To add a new subnet to an existing adapter, click *New*.
- To delete a subnet, select a subnet, then click *Delete*. More than one must be configured for you to delete a subnet.
- To modify an existing subnet, click the IP address of the subnet.

4 To configure a new subnet or a new IP address for a subnet, configure the following fields:



Adapter eth0

Subnet: 10.10.15.0

Subnet Mask: * 255.255.252.0

IP Address List *

☐ IP Addresses
☐ 10.10.16.60

Server(s) must be updated before changes made on this panel will be used.

Subnet: Displays the address of the subnet that you are modifying. This is empty if you are creating a new one.

Subnet Mask: (Required) Specifies the subnet mask address for this subnet. The address can be specified in standard dotted format or in CIDR format

IP Addresses: Allows you to manage the IP addresses assigned to the subnet.

- To add an address, click *New*, specify the address, then click *OK*.
- To delete an address, select the address, then click *Delete*.
- To change the IP address, see Section 4.3, "Changing the IP Address of the Access Gateway," on page 46.

5 Click *OK*.

6 Configure the *Adapter List Options*.

These options let you change settings for the network adapters on the Access Gateway to ensure compatibility with an existing LAN. Modify the default settings only if your LAN requires specialized adapter card changes.

- **Speed:** Select *Default*, *10 MB*, *100 MB*, or *1000 MB*.
- **Duplex:** Select *Default*, *Half*, or *Full*.

  IMPORTANT: Some network adapter drivers do not correctly detect duplex settings. This is a general industry problem with Fast Ethernet technology.

  If your Access Gateway isn't performing as expected, check to ensure that the duplex settings for its network adapters match your network configuration. It might be necessary to manually configure the duplex settings on both your Access Gateway and your Ethernet switch or hub.
- **NAT:** Select *Dynamic* or *Disabled*.

  If the Access Gateway is serving as a router, and your network employs non-unique private IP addresses, you can configure the Access Gateway to provide Network Address Translation (NAT) services.

  For example, if you have a 10.0.0.0 private network on eth0 and a registered public network such as 130.0.0.0 on eth1, the clients on the private network can access the Internet through the Access Gateway, provided that the *Dynamic* option is selected in the NAT drop-down list for the eth1 adapter.

The Access Gateway then functions as a network address translator and dynamically maps the private, non-routable 10-net addresses to the registered public address assigned to eth1.

IMPORTANT: You cannot configure a reverse proxy on an IP address assigned to an adapter that has the *Dynamic* option set for NAT. NAT and a reverse proxy cannot coexist on the same adapter.

Custom load parameters: (NetWare only) Allows you to specify non-standard load parameters for a custom driver. If you used the custom driver option during installation and the documentation for this driver specified some custom load parameters, enter these parameters in the text box.

7. To save your changes to browser cache, click *OK*.
8. On the Server Configuration page, click *OK*, then click *Update* > *OK*.

15.7.2 Viewing and Modifying Gateway Settings

The gateway settings display the current gateway configuration that the Access Gateway is using to route packets. From this page, you can also to configure additional gateways. During installation, you could specify only a default gateway. You must have at least one gateway defined for the Access Gateway to function.

The Access Gateway routes requests to specific destinations through these gateways. If a request could be routed through multiple gateways, the Access Gateway chooses the gateway associated with the most restrictive mask (the smallest range of destination addresses). The default gateway is used only when no other routes apply.

Gateways fall within the following three basic groups:

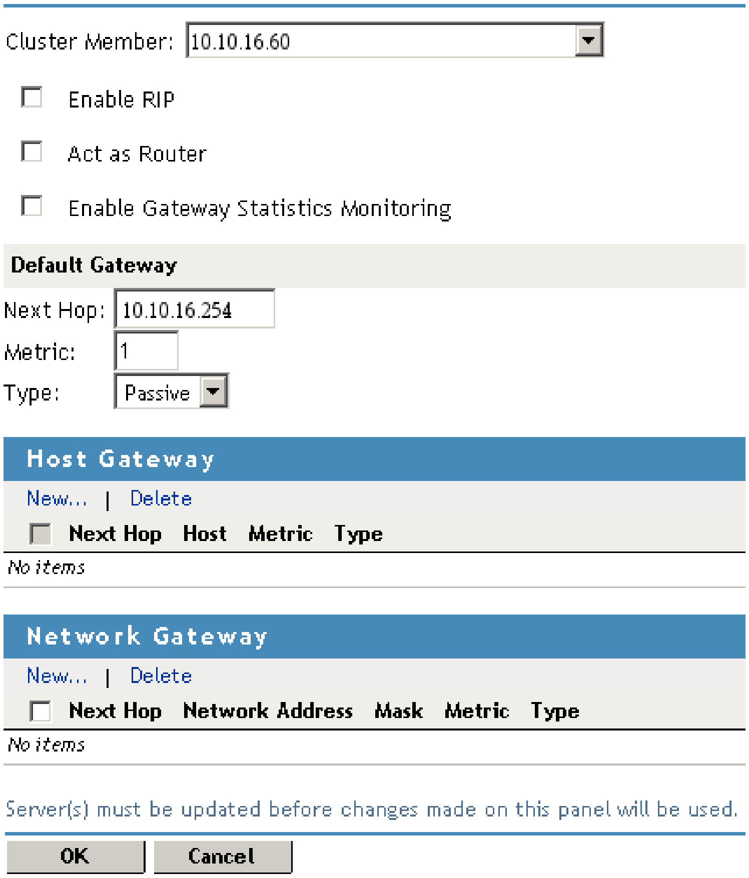
- Host gateways for specific destination addresses.
- Network gateways for destination addresses that fall within specific subnets.
- The default gateway for destination addresses that aren’t covered by host or network gateways.

The Access Gateway uses additional gateways only when the *Act As Router* option is selected. When this option is selected, you can add Host Gateways and Network Gateways. When configuring a Host Gateway or Network Gateway, you specify the IP address of the host or network gateway in the *Next Hop* field. This address must be on the same subnetwork as the IP address for the Access Gateway.

IMPORTANT: If you enter an IP address that is on a different subnetwork, the Linux Access Gateway reports this error on the Health page, after the configuration has been applied. The NetWare Access Gateway ignores the configuration error and does not report it.

To modify your current gateway configuration:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Gateways*.



Cluster Member: 10.10.16.60

☐ Enable RIP

☐ Act as Router

☐ Enable Gateway Statistics Monitoring

Default Gateway

Next Hop: 10.10.16.254

Metric: 1

Type: Passive

Host Gateway

☐	Next Hop	Host	Metric	Type

No items

Network Gateway

☐	Next Hop	Network Address	Mask	Metric	Type

No items

Server(s) must be updated before changes made on this panel will be used.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the *Cluster Member* field. All changes made to this page apply to the selected server.

3 Fill in the following fields:

Enable RIP: (NetWare only) Allows you to turn on the Routing Information Protocol 1. Through this protocol, the Access Gateway is able to learn routes.

Act as Router: Select this option if the Access Gateway functions as the default gateway for clients on the network. If you select this option, you can specify additional gateways.

Enable Gateway Statistics Monitoring: Select this option if you want to gather statistics and monitor the traffic on the gateways.

4 Configure your default gateway, which specifies the gateway to use when no other routes apply. Configure the following:

Next Hop: The IP address of the gateway.

Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

Type: Gateways are active if they publish their presence, or passive if they do not.

5 Configure your host gateways, which are the gateways to be used for packets being sent to specific hosts. When you select *New* from the *Host Gateway* list, you are asked for the following information:

Next Hop: The address of the host gateway that is to be used.

Host: The IP address of the destination host. Valid addresses cannot be the first or last address of a class and must be unique.

Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

Type: Gateways are active if they publish their presence, or passive if they do not.

Click *OK* when the fields are configured.

6. Configure your network gateways, which are the gateways to be used for packets being sent to specific subnets. When you select *New* from the *Network Gateway* list, you are asked for the following information:

   Next Hop: The address of the gateway that is to be used.

   Network Address: The subnet address for the destination IP address range. You can also enter a specific IP address on a given subnet, and the Access Gateway calculates the subnet address using the mask.

   Mask: The subnet mask for the subnet or IP address above. A valid entry must be at least as large as a class mask where a Class A mask is 255.0.0.0, a Class B mask is 255.255.0.0, and Class C, D, and E masks are 255.255.255.0.

   Metric: A relative number indicating the bias you can add to the normal flow of gateway logic. Specifying a number higher than 1 makes this resource more expensive and alters the gateway logic used. Valid numbers include 1 through 16.

   Type: Gateways are active if they publish their presence, or passive if they do not.

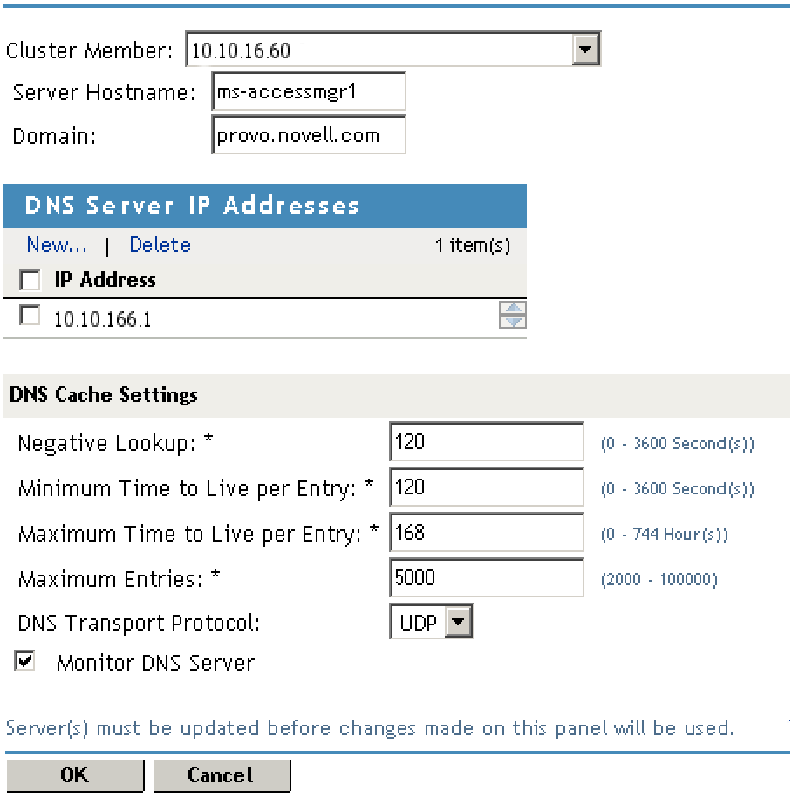
   Click *OK* when the fields are configured.

7. To save your changes to browser cache, click *OK*.
8. On the Server Configuration page, click *OK*, then click *Update* > *OK*.

15.7.3 Viewing and Modifying DNS Settings

The DNS page displays the current configuration for domain name services and allows you to modify it.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *DNS*.



Cluster Member: 10.10.16.60

Server Hostname: ms-accessmgr1

Domain: provo.novell.com

DNS Server IP Addresses

☐ IP Address
☐ 10.10.166.1

DNS Cache Settings

Negative Lookup: * 120 (0 - 3600 Second(s))

Minimum Time to Live per Entry: * 120 (0 - 3600 Second(s))

Maximum Time to Live per Entry: * 168 (0 - 744 Hour(s))

Maximum Entries: * 5000 (2000 - 100000)

DNS Transport Protocol: UDP

☑ Monitor DNS Server

Server(s) must be updated before changes made on this panel will be used.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the *Cluster Member* field. All changes made to this page apply to the selected server.

3 Fill in the following fields:

Server Hostname: Displays the unique host or computer name that you have assigned to the Access Gateway machine. If you modify this name, you need to modify the entry for the Access Gateway in your DNS server to resolve this new name.

Domain: Specifies the domain name for your network. Your DNS server must be configured to resolve the combination of the server hostname and the domain name to the Access Gateway machine. This field assumes you are using dotted names for your machines, such as sales.mytest.com, where sales is the *Server Hostname* and mytest.com is the *Domain*.

DNS Server IP Addresses: Displays the IP addresses of the servers on your network that resolve DNS names to IP addresses. You can have up to three servers in the list. If you specified any addresses during installation, they appear in this list. To manage the servers in this list, select one of the following options:

- **New:** To add a server to the list, click this option and specify the IP address of a DNS server.
- **Delete:** To delete a server from the list, select the address of a server, then click this option.
- **Order:** To modify the order in which the DNS servers are listed, select the server, then click either the up-arrow or the down-arrow buttons. The first server in the list is the first server contacted when a DNS name needs to be resolved.

4 Configure the DNS Cache Settings. These options allow you to control the refresh of DNS information. These are all standard DNS options.

Negative Lookup: Specifies how long a failed DNS lookup domain name remains in cache. If the Access Gateway cannot resolve a domain name, it stores that information in its cache for the specified amount of time. If the Access Gateway receives requests for that domain name within this period, it sends a "Bad Gateway" error message to the browser and does not resolve the domain name again. Valid field values include 0–3600 seconds. The default is120 seconds.

Minimum Time To Live per Entry: Specifies the minimum amount of time that DNS entries remain in cache before they expire. This is the minimum value the Access Gateway uses regardless of the value the DNS server returns. Valid field values include 0–3600 seconds. The default is 120 seconds.

Maximum Time To Live per Entry: Specifies the maximum amount of time that DNS entries remain in cache before they expire. This is the maximum value the Access Gateway uses regardless of the value the DNS server returns. Valid field values include 0–744 hours. The default is 168 hours.

Maximum Entries: Specifies the maximum number of DNS cache entries. When this number is reached, the Access Gateway deletes old entries to make room for newer ones. Valid field values include 2000–100000. The default is 5000.

DNS Transport Protocol: Specifies the transport protocol that DNS uses on the network where the Access Gateway is installed. Valid values are UDP and TCP. The default is UDP.

Monitor DNS Server: (NetWare only) If selected, allows the Access Gateway to monitor DNS server availability by pinging the configured servers every minute. This ensures timely handling of DNS requests. You should deselect this item if the Access Gateway accesses DNS through a connection that is not kept continually open, such as a dial-up phone line or ISDN connection.

Keep in mind, however, that deselecting this option causes the DNS configuration on the Health tab to display the following message: `"(Passed) Domain and DNS Servers configured"`. When this option is enable, the Health tab displays the following message: `"(Passed) Domain and DNS Servers configured and active"`.

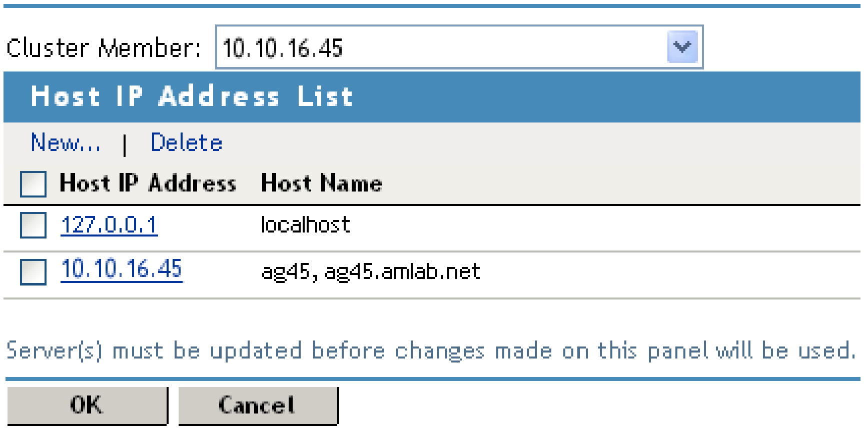
5 To save your changes to browser cache, click *OK*.

6 On the Server Configuration page, click *OK*, then click *Update* > *OK*.

15.7.4 Configuring Hosts

(Linux only) You can configure the Linux Access Gateway to have multiple host names.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Hosts*.



Cluster Member: 10.10.16.45

Host IP Address List

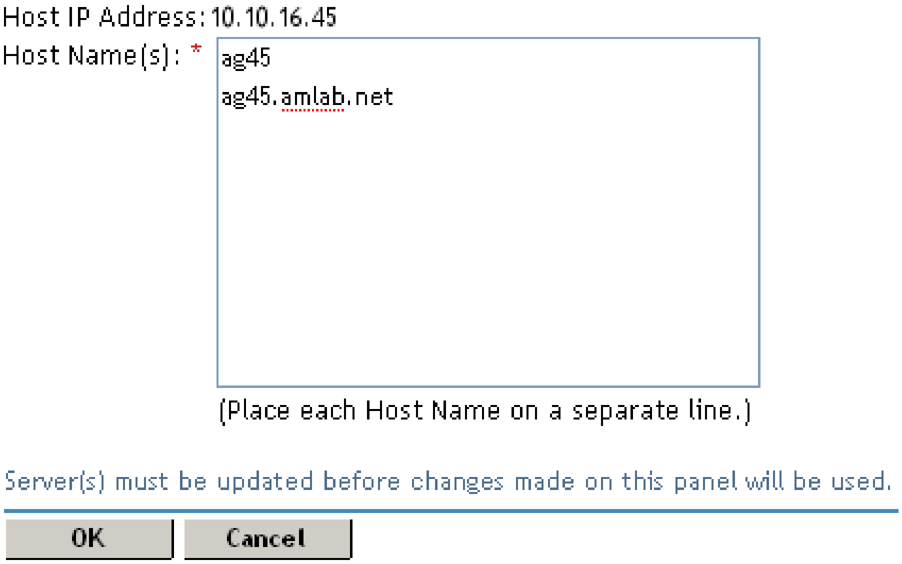
☐ Host IP Address	Host Name
☐ 127.0.0.1	localhost
☐ 10.10.16.45	ag45, ag45.amlab.net

Server(s) must be updated before changes made on this panel will be used.

This page displays a list of host IP addresses.

2 (Conditional) If the Access Gateway is a member of a cluster, select the server you want to configure from the list of servers in the *Cluster Member* field. All changes made to this page apply to the selected server.

3 To add a new host name to an existing IP address, click the name of a *Host IP Address*.



Host IP Address: 10.10.16.45

Host Name(s): * ag45
ag45.amlab.net

(Place each Host Name on a separate line.)

Server(s) must be updated before changes made on this panel will be used.

4 In the *Host Name(s)* text box, specify a name for the host. Place each host name on a separate line. Then click *OK*.

5 To add a new IP address and host name, click *New* in the *Host IP Address List* section, then specify the IP address. In the *Host Name(s)* text box, specify a host name, then click *OK*.

6 To delete a host, select the check box next to the host you want to delete, then click *Delete*.

7 To save your changes to browser cache, click *OK*.

8 On the Server Configuration page, click *OK*, then click *Update* > *OK*.

15.7.5 Adding New Network Interfaces to the Linux Access Gateway

If you add new network interface cards to the Linux Access Gateway machine after installation, you need to scan for these cards. Then you can configure them.

1. In Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Server]*.
2. Click *New NIC* to scan for new network interface, then click *OK* to confirm.

   You can click the *Command Status* tab to check if the scan has completed.
3. Click *Access Gateways*, then click *Edit* for the cluster or server that has the new card.
4. Click *Adapter List*. If the server is a member of a cluster, select the cluster member you want to configure.

   The newly added network interface is displayed here.
5. In the newly added adapter section, click *New*, then configure the subnet mask and IP address.
6. To save your changes to browser cache, click *OK*.
7. On the Server Configuration page, click *OK*, then click *Update* > *OK*.

15.8 Customizing Log Out

If any of your protected resources have a logout page or button, you need to redirect the user's logout request to the Access Gateway logout page. The Access Gateway can then clear the user's session and log the user out of any other resources that have been enabled for single sign-on. If you do not redirect the user's logout request, the user is logged out of one resource, but the user's session remains active until inactivity closes the session. If the user accesses the resource again before the session is closed, single sign-on re-authenticates the user to the resource, and it appears that the logout did nothing.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxy / Authentication*.
2. In the *Embedded Service Provider* section, view the path to the AGLogout page in the *Logout URL* option.

   The Logout URL displays the URL that you need to use for logging users out of protected resources. This option is not displayed until you have created at least one reverse proxy with a proxy service. If you create two or more reverse proxies, you can select which one is used for authentication, and the logout URL changes to match the assigned reverse proxy. For more information on changing the authentication proxy, see Section 17.3.2, "Changing the Authentication Proxy Service," on page 306.
3. Use this path to redirect application logout requests to this page.
4. Click *OK*.

For backwards compatibility, the Linux Access Gateway currently supports the following logout pages:

- /cmd/BM-Logout
- /cmd/ICSLogout

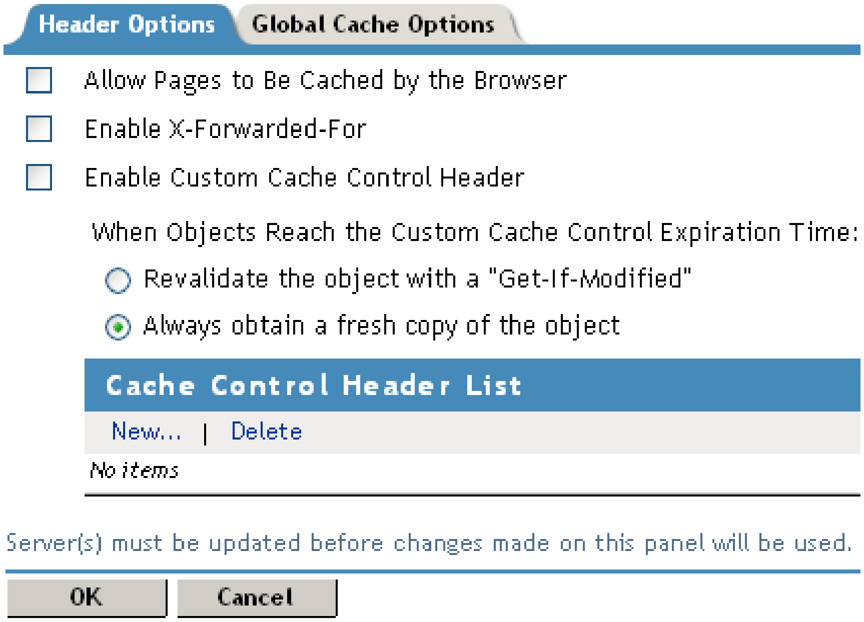
These pages have been disabled on the NetWare Access Gateway, and in a future release, will be disabled on the Linux Access Gateway. If you have applications that use these pages for redirecting the user's logout request, we suggest that you update them to use the AGLogout page. The AGLogout page does a global logout, logging the user out of all resources, Access Gateways, Identity Servers, and service providers.

15.9 Configuring X-Forwarded-For Headers

X-Forwarded-For headers are used to pass browser ID information along with browser request packets. If the headers are included, Web servers can determine the origin of browser requests they receive. If the headers are not included, browser requests have anonymity.

Deciding whether to enable X-Forwared-For headers requires that you weigh the desires of browser users to remain anonymous against the desires of Web server owners (e-commerce sites, for example) to collect data about who is accessing their sites. This option is disabled by default. To enable it:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *HTTP Options* > *Header Options*.



☐ Allow Pages to Be Cached by the Browser

☐ Enable X-Forwarded-For

☐ Enable Custom Cache Control Header

When Objects Reach the Custom Cache Control Expiration Time:

○ Revalidate the object with a "Get-If-Modified"

● Always obtain a fresh copy of the object

No items

Server(s) must be updated before changes made on this panel will be used.

2. Select the *Enable X-Forwarded-For* option.

   With this option selected, the proxy service either adds information to an existing X-Forwarded-For or Forwarded-For header, or creates a header if one doesn't already exist. Leaving the option deselected causes the proxy service to remove X-Forwarded-For headers from any Web requests passing through the proxy service.
3. To save your changes to browser cache, click *OK*.
4. To apply your changes, click the *Access Gateways* link, then click *Update* > *OK*.

15.10 Upgrading the Access Gateway Software

You can upgrade the software currently running on Access Gateway to a newer version without losing configuration information and with down time limited to the time it takes the Access Gateway to restart. See "Upgrading the Linux Access Gateway" and "Upgrading the NetWare Access Gateway" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

15.11 Exporting and Importing an Access Gateway Configuration

You can export an existing Access Gateway configuration as well as its dependent policies, and then import this configuration to a new machine. This feature is especially useful for deployments that set up configurations in a staging environment, test and validate the configuration, then want to deploy the configuration on new hardware that exists in the production environment.

IMPORTANT: The export feature is not a backup tool. The export feature is designed to handle configuration information applicable to all members of a cluster, and network IP addresses and DNS names are filtered out during the import. (The server-specific information that is filtered out is the information you set specifically for each member in a cluster.) If you want a copy of all configuration information, including server-specific information, you need to perform a backup. See Chapter 2, "Backing Up and Restoring Components," on page 31.

When exporting the file, you can select to password protect the file, which encrypts the file. If you are using the exported file to move an Access Gateway from a staging area to a production area and you need to change the names of the proxy services and DNS names from a staging name to a production name, do not select to encrypt the file. You need a simple text file so you can search and replace these names. If you select not to encrypt the file, remember that the file contains sensitive information and protect it accordingly.

The following sections explain this process:

- Section 15.11.1, "Exporting the Configuration," on page 274
- Section 15.11.2, "Importing the Configuration," on page 276
- Section 15.11.3, "Cleaning Up and Verifying the Configuration," on page 276

15.11.1 Exporting the Configuration

1 In the Administration Console, click *Access Manager* > *Access Gateway* > *[Name of Access Gateway]*.

2 Click *Configuration* > *Export*.

3 (Conditional) If you want to encrypt the file, fill in the following fields:

   Password protect: Select this option to encrypt the file.

   Password: Specify a password to use for encrypting the file. When importing the configuration onto another device, you are prompted for this password.

4 Click *OK*, then select to save the configuration to a file.

   The filename is the name of the Access Gateway with an `xml` extension.

5 (Conditional) If you want to change the names of the proxy services and their DNS names from a staging name to a production name, complete the following:

   5a Open the file in a text editor.

   5b Search and remove the staging suffix.

If you have specified DNS names with a staging suffix (for example, `innerwebstaging.provo.novell.com`), you can search for `staging.provo.novell.com` and remove `staging` from the name.

In particular, you need to change the following:

- Any fully qualified DNS names from the staging name to the production name (DNSName elements in the file).
- The cookie domains associated with each proxy service (AuthenticationCookieDomain elements in the file)
- The URL masks in Pin Lists that contain fully qualified names (URLMask elements in the file.

Depending upon your naming standards, you might want to change the names of the following:

- UserInterfaceID elements (proxy service, pin list, and protected resource user interface ID's)
- Description elements (proxy service, pin list, and protected resource descriptions)
- Name (proxy service, pin list, and protected resource names)
- SubServiceID elements
- MultiHomeMasterSubserviceIDRef elements
- LogDirectoryName elements
- ProfileIDRef elements
- ProtectedResourceID elements
- ProfileID elements (TCP Listen options name)

5c (Conditional) If your Web servers in the staging area have different IP addresses and host names than the Web Servers in the production area, you can search and replace them in the configuration file or wait until after the import and modify them in the Administration Console.

6 Export the policies used by the Access Gateway. In the Administration Console, click *Access Manager* > *Policies*, then either select *Name* to include all policies or individually select the policies to export.

You need to export all Access Gateway policies and any Role policies used by the Access Gateway policies.

7 Click *Export* and modify the proposed filename if needed.

8 Click *OK*, then select to save the policy configurations to a file.

9 (Conditional) If you have created multiple policy containers, select the next policy container in the list, and repeat Step 6 through Step 8.

The policies for each container must be saved to a separate export file.

10 (Conditional) If your policies redirect users to staging URLs when they are denied access, search and replace these URLs with the production URLs. Open the policy file with a text editor and search for your staging name.

11 Copy the Access Gateway and policy configuration files to a place accessible by the new Access Gateway.

12 Continue with Section 15.11.2, “Importing the Configuration,” on page 276.

15.11.2 Importing the Configuration

1. Verify that the Access Gateway meets the conditions for an import:
    - The Access Gateway should not be a member of a cluster. If it is a member of a cluster, remove it from the cluster before continuing.

      In the Administration Console, click *Access Manager* > *Access Gateways*, select the Access Gateway, then click *Actions* > *Remove from Cluster*.

      You can create a cluster and add this machine to the cluster as the primary server after you have completed the import.
    - The Access Gateway should be an unconfigured machine. If it contains reverse proxies, delete them before continuing.

      In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxies / Authentication*. In the *Reverse Proxy List*, select *Name*, then click *Delete*. Update the Access Gateway and the Identity Server.
2. In the Administration Console, click *Access Manager* > *Policies*.

   The policies that the Access Gateway is dependent upon must be imported first.
3. (Conditional) If you have exported policies from more than one container, create the policy containers. Click the *Manage Policy Containers* icon; in the *Container List*, click *New*, specify the name for the container, then click *OK*.
4. (Conditional) If your system already contains policies, delete them if they aren't being used.

   If they are in use and you have policies with the same names as the policies you are going to import, you need to manually reconcile the duplicate policies. See Step 5 in Section 15.11.3, "Cleaning Up and Verifying the Configuration," on page 276.
5. In the Policy List, click *Import*.
6. Browse to the location of the policy configuration file, select the file, then click *OK*.
7. (Conditional) If you exported multiple policy configuration files, repeat Step 5 and Step 6.
8. Enable all new Role policies. Click *Identity Servers* > *Edit* > *Roles*.
9. Either select *Name* to enable all policies or individually select the policies, then click *Enable*.
10. Click *OK*, then click *Update*.
11. To import the Access Gateway configuration, click *Access Gateways* > *[Name of Access Gateway]* > *Configuration* > *Import*.
12. Browse to the location of the file, select the file, enter a password if you specified one on export, then click *OK*.
13. Continue with Section 15.11.3, "Cleaning Up and Verifying the Configuration," on page 276.

15.11.3 Cleaning Up and Verifying the Configuration

1. When the configuration import has finished, verify the configuration for your reverse proxies.

   1a Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.

   1b Verify the listening address.

   This is especially important if your Access Gateway has multiple network adapters. By default, the IP address of eth0 is always selected as the listening address.

   1c Verify the certificates assigned to the reverse proxy.

The Subject Name of the certificate should match the published DNS name of the primary proxy service in the *Proxy Service List*.

1d Verify the Web Server configuration. In the *Proxy Service List*, click the *Web Server Addresses* link. Check the following values:

- **Web Server Host Name.** If this name has a staging prefix or suffix, remove it.
- **IP addresses in the Web Server List.** If the IP addresses in the production area are different from the IP addresses in the staging area, modify the IP addresses to match the production area.
- **Certificates.** If you have configured SSL or mutual SSL between the proxy service and the Web servers, configure the *Web Server Trusted Root* and *SSL Mutual Certificate* options. The export and import configuration option does not export and import certificates.

1e Click *OK* twice.

2. (Conditional) If you have multiple reverse proxies, repeat Step 1 for each proxy service.
3. On the Configuration page, click *Reverse Proxy / Authentication*, then select the *Identity Server Cluster* configuration.
4. If you have multiple reverse proxies, verify that the Reverse Proxy value in the *Embedded Service Provider* section is the reverse proxy you want to use for authentication, then click *OK* twice.
5. (Conditional) If the Administration Console already contained some policies, verify that you do not have policies with duplicate names. Click *Access Manager* > *Policies*.

   Policies with duplicate names have Copy-*n* appended to the end of the name, with *n* representing a number. If you have duplicates, reconcile them:

   - If they contain the same rules, you need to reconfigure the resources using one policy to use the other policy before you can delete the duplicate policy.
   - If they contain different rules, rename the duplicate policies.
6. (Conditional) Apply any policy configuration changes.
7. Click *Access Gateways* > *Update*.
8. Click *Identity Servers* > *Update*.
9. (Optional) Create a cluster configuration and add this server as the primary server.

Configuring the Cache Settings

16

One of the major benefits of using an Access Gateway to protect Web resources is that it can cache the requested information and send it directly to the client browser rather than contacting the origin Web resource and waiting for the requested information to be sent. This can significantly accelerate access to the information.

The object cache on an Access Gateway is quite different from a browser's cache, which all users access when they click the *Back* button and which can serve stale content that doesn't accurately reflect the fresh content on the origin Web server.

The Access Gateway caching system uses a number of methods to ensure cache freshness. Most time-sensitive Web content is flagged by Webmasters in such a way that it cannot become stale unless a caching system ignores the Webmaster's settings. The Access Gateway honors all flags that affect cache freshness, including Time to Expire, Don't Cache, and Must Revalidate directives.

In addition, the Access Gateway can be fine-tuned for cache freshness in the following ways:

- Accelerated checking of objects that have longer than desirable Time to Expire headers
- Delayed checking of objects that have shorter than desirable Time to Expire headers
- Checking for freshness of objects that do not include Time to Expire headers

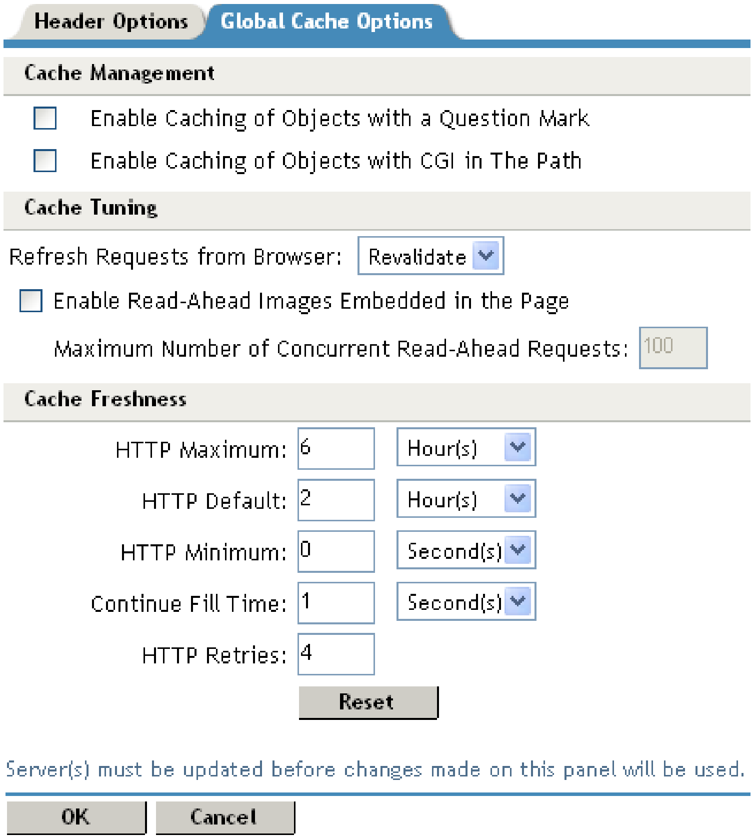
The following sections describe the features available to fine-tune this process for your network:

- Section 16.1, "Configuring Global Caching Options," on page 279
- Section 16.2, "Controlling Browser Caching," on page 282
- Section 16.3, "Configuring Custom Cache Control Headers," on page 283
- Section 16.4, "Configuring a Pin List," on page 285
- Section 16.5, "Configuring a Purge List," on page 289
- Section 16.6, "Purging Cached Content," on page 290
- Section 16.7, "Preventing a Web Site from Being Cached," on page 290

16.1 Configuring Global Caching Options

Caching is configured at the proxy service level. This gives you a great deal of control in specifying what you want cached.

1. Click *Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > HTTP Options > Global Cache Options*.



Cache Management

☐ Enable Caching of Objects with a Question Mark

☐ Enable Caching of Objects with CGI in The Path

Cache Tuning

Refresh Requests from Browser: Revalidate

☐ Enable Read-Ahead Images Embedded in the Page

Maximum Number of Concurrent Read-Ahead Requests: 100

Cache Freshness

HTTP Maximum: 6 Hour(s)

HTTP Default: 2 Hour(s)

HTTP Minimum: 0 Second(s)

Continue Fill Time: 1 Second(s)

HTTP Retries: 4

Server(s) must be updated before changes made on this panel will be used.

2 Configure the *Cache Management* options:

Enable Caching of Objects with a Question Mark: If this option is selected, a cacheable object is cached if it has a question mark in the URL.

Enable Caching of Objects with CGI in the Path: If this option is selected, a cacheable object is cached if it has `/cgi` in its URL.

Objects that meet these criteria are only cached if they are also cacheable objects. Web server administrators can mark objects as non-cacheable. When so marked, these objects are not cached, even when the above options are selected.

If you disable both of these options, it does not mean that objects with question marks or cgi in their paths cannot be cached. These objects can match some other criteria and be cached.

3 Configure the *Cache Tuning* options.

These options restrict or enable functionality that affects all the resources protected by a proxy service.

Refresh Requests from Browser: When a user clicks *Refresh* or *Reload* in the browser, this action sends a new request to the Web server. Select one of the following options to control how the proxy service handles the request:

- **Refill:** Causes the proxy service to send the request to the Web server
- **Revalidate:** (Linux only) Causes the proxy service to check whether the current information is valid. If it is, the currently cached information is returned. If it isn’t valid, the request is forwarded to the Web server.

- **Ignore:** Causes the proxy service to ignore the request and send the data from cache without checking to see if the cached data is valid.

Enable Read-Ahead Images Embedded in the Page: If this option is selected, the proxy service retrieves and caches objects that have been flagged Read-Ahead. You specify the maximum number of read-ahead objects the proxy service retrieves in the *Maximum Number of Concurrent Read-Ahead Requests* field.

Maximum Number of Concurrent Read-Ahead Requests: Sets a limit on the number of read-ahead images that can be cached.

4. (Optional) Modify the Cache Freshness settings. Use the *Reset* button to return these settings to their default values.

   These options govern when the proxy service revalidates requested cached objects against those on their respective origin Web servers. If the objects have changed, the proxy service re-caches them.

   HTTP Maximum: Specifies the maximum time the proxy service serves HTTP data from cache before revalidating it against content on the origin Web server. No object is served from cache after this value expires without being revalidated.

   This overrides a freshness or Time to Expire directive specified by the Webmaster if he or she specified a longer time.

   You use this value to reduce the maximum time the proxy service waits before checking whether requested objects need to be refreshed. The default is 6 hours.

   HTTP Default: Specifies the maximum time the proxy service serves HTTP data for which Webmasters have not specified a freshness or Time to Expire directive. The default is 2 hours.

   HTTP Minimum: Specifies the minimum time the proxy service serves HTTP data from cache before revalidating it against content on the origin Web server. No requested object is revalidated sooner than specified by this value.

   This overrides the freshness or Time to Expire directive specified by the Webmaster if he or she specified a shorter time.

   You can use this value to increase the minimum time the proxy service waits before checking whether requested objects need to be refreshed. This parameter does not override No Cache or Must Revalidate directives from the origin Web server.

   The default value is 0, which allows the proxy service to honor the Time To Expire directive of each object (unless it is longer than the *HTTM Maximum* option). If the *HTTP Minimum* option is set to a value other than 0, the value overrides any object’s Time to Expire directive that is shorter than the value set. The default is 0.

   Continue Fill Time: Specifies the how long the proxy service ignores browser request cancellations and continues downloading objects from the target Web server until the download is complete. The default is 1 second.

   HTTP Retries: Specifies the number of retry requests to issue to a Web server. The default is 4 retries.

5. To save your changes to browser cache, click *OK*.
6. To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

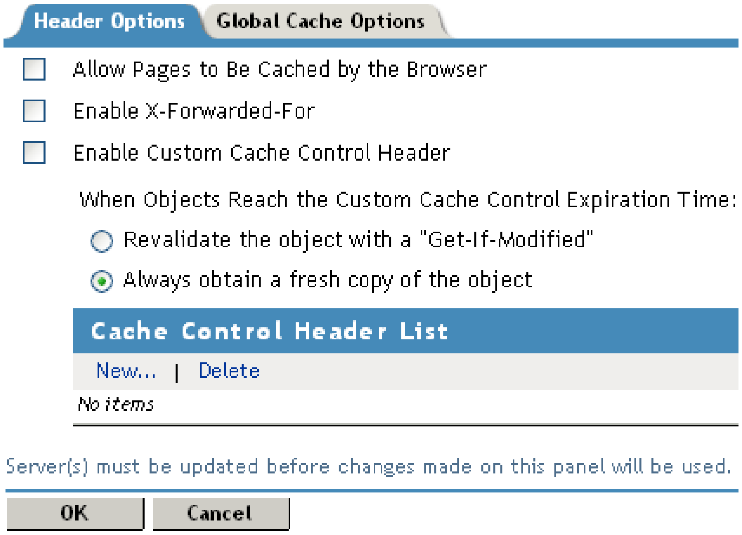
16.2 Controlling Browser Caching

Webmasters control how browsers cache information by adding the following cache-control directives to the HTTP headers:

```
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: private
Cache-Control: public
Pragma: no-cache
```

You can configure how the proxy service responds to these directives in the HTTP header.

1. In the Administration Console, click *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > HTTP Options > Header Options*.



☐ Allow Pages to Be Cached by the Browser

☐ Enable X-Forwarded-For

☐ Enable Custom Cache Control Header

When Objects Reach the Custom Cache Control Expiration Time:

○ Revalidate the object with a "Get-If-Modified"

● Always obtain a fresh copy of the object

Cache Control Header List

New... | Delete

No items

Server(s) must be updated before changes made on this panel will be used.

2. To mark all pages coming through this host as cacheable on the browser, select *Allow Pages to be Cached by the Browser*.

   When this option is enabled, the no-cache and no-store headers are not injected into the HTTP header.

   You need to select this option if you have a back-end application that updates the data in the Last-Modified or ETag HTTP headers. These changes are forwarded from the Web server to the browser only when this option is enabled.

   You need to select this option is you want the Expires HTTP header forwarded from the Web server to the browser.

   If this option is not selected, all pages are marked as non-cacheable on the browser. This forces the browser to request a resend of the data from the Access Gateway when a user returns to a previously viewed page.

3. To configure custom caching instructions, see Section 16.3, "Configuring Custom Cache Control Headers," on page 283.
4. To save your changes to browser cache, click *OK*.
5. To apply the changes, click the *Access Gateways* link, then click *Update > OK*.

16.3 Configuring Custom Cache Control Headers

In addition to fine-tuning cache freshness by using the global HTTP timers, as explained in Section 16.1, "Configuring Global Caching Options," on page 279, you can configure each proxy service to recognize custom headers in HTTP packets. Your Web server can then use these headers for transmitting caching instructions that only the Access Gateway can recognize and follow.

- Section 16.3.1, "Understanding How Custom Cache Control Headers Work," on page 283
- Section 16.3.2, "Enabling Custom Cache Control Headers," on page 284

16.3.1 Understanding How Custom Cache Control Headers Work

Only the proxy service containing the custom header definition follows the cache policies specified in the custom headers.

All other proxy services, requesting browsers, and external proxy caches (transparent caches, client accelerators, etc.), do not recognize the custom headers. They follow only the cache policies specified by the standard cache control headers.

This means that you have the following options for configuring your Web server:

- You can specify that browsers and/or external caches cannot cache the objects, but the proxy service can.

  This lets you offload request-processing from the origin Web server while still requiring that users return to the site each time they request an object.
- You can also specify separate cache times for browsers, external caches, and the proxy service.

To implement custom cache control headers, you must do the following:

- Configure a proxy service to use custom cache control headers by enabling the feature and specifying a header string such as MYCACHE (see Section 16.3.2, "Enabling Custom Cache Control Headers," on page 284).
- Configure the Web servers of the proxy service to send an HTTP header containing the defined string and the time in seconds that the object should be retained in cache (for example, MYCACHE: 60).

  If the number is non-zero, the Access Gateway treats the reply as if it has the following headers:

  ```
  Cache-Control: public
  Cache-Control: max-age=number
  ```

  If the number is zero (0), the Access Gateway treats the reply as if it has the following header:

  ```
  Cache-Control: no-cache
  ```
- Ensure that the Web server continues to send standard HTTP cache-control headers so that browsers and external caches follow the caching policies you intend them to.

For example, you can configure the following:

- Use an Expires or Cache-Control: Max-Age header to specify that browsers should cache an object for two minutes.
- Use a Cache-Control: Private header to prevent external caches from caching the object at all.

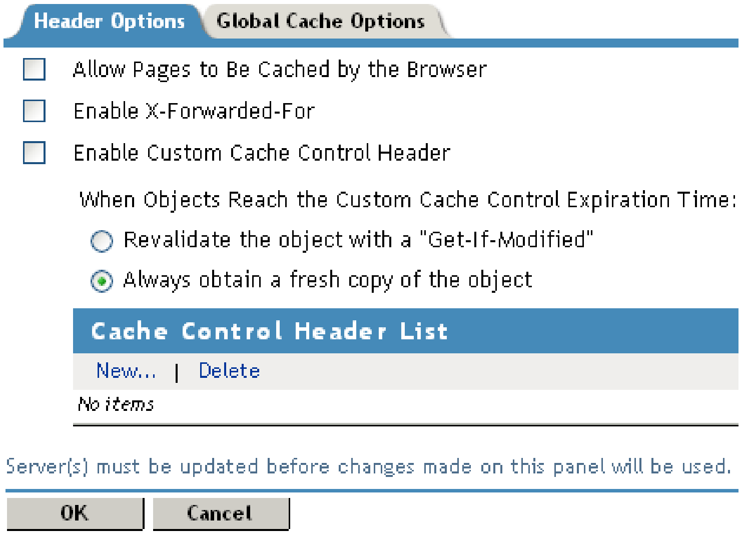
- Use a custom cache control header, such as MYCACHE: 1800, to indicate that the proxy service should cache the object for 30 minutes.

Custom Cache Control Headers override the following standard HTTP cache-control headers on the Access Gateway, but they do not affect how browsers and external caches respond to them:

```
Cache-Control: no-store
Cache-Control: no-cache
Cache-Control: max-age=number
Cache-Control: private
Cache-Control: public
Pragma: no-cache
Expires: date
```

16.3.2 Enabling Custom Cache Control Headers

1 In the Administration Console, click *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > HTTP Options > Header Options*.



☐ Allow Pages to Be Cached by the Browser
☐ Enable X-Forwarded-For
☐ Enable Custom Cache Control Header

When Objects Reach the Custom Cache Control Expiration Time:
○ Revalidate the object with a "Get-If-Modified"
● Always obtain a fresh copy of the object

No items

Server(s) must be updated before changes made on this panel will be used.

2 To enable the use of custom headers, select *Enable Custom Cache Control Header*.

With this option selected, the proxy service searches HTTP packets for custom cache control headers, and caches the objects according to its policies. The policy contains a timer, which specifies how long the object can be cached before checking with the Web server for updates.

3 Select one of the following options to specify what occurs when the custom cache control expiration time expires.

- **Revalidate the object with a “Get-If-Modified”:** Causes the proxy service to update the object in cache only if the object has been modified.
- **Always obtain a fresh copy of the object:** Causes the proxy service to update the object in cache, even if the object has not been modified.

4 In the *Cache Control Header List*, select *New* and specify a name for the header, for example MYCACHE.

5 To save your changes to browser cache, click *OK*.

6 To apply the changes, click the *Access Gateways* link, then click *Update > OK*.

7. Modify the pages on the Web server that you want to the set custom caching intervals for the Access Gateway. To the HTTP header, add a string similar to the following:

   `MYCACHE:600`

   The numeric value indicates the number of seconds the Access Gateway can retain the object in cache. A value of zero prevents the Access Gateway from caching the object. This cache interval can be different than the value set for browsers (see Section 16.3.1, "Understanding How Custom Cache Control Headers Work," on page 283).

8. Ensure that the Web server continues to send the following standard HTTP cache-control headers:
   - Cache-Control: Max-Age headers that cause browsers to cache object for no longer than two minutes.
   - Cache-Control: Private headers that cause external caches to not cache the objects.

When your Web server sends an object with the MYCACHE header in response to a request made through the Access Gateway, the proxy service recognizes the custom header and caches the object for 10 minutes. Requesting browsers cache the object for only two minutes, and external caches do not cache the object.

Thus, the Access Gateway off-loads a processing burden from the Web server by caching the frequently requested objects for 10 minutes (the value you specified in Step 7). Browsers, on the other hand, must always access the Access Gateway to get the objects if their previous requests are older than two minutes. And the objects in the cache of the Access Gateway are kept fresh due to their relatively brief time-to-live value.

16.4 Configuring a Pin List

A pin list contains URL patterns for identifying objects on the Web. The Access Gateway uses the list to prepopulate the cache, before any requests have come in for the content. This accelerates user access to the content because it is retrieved from a local cache rather than from an exchange with the Web server, which would read it from disk.

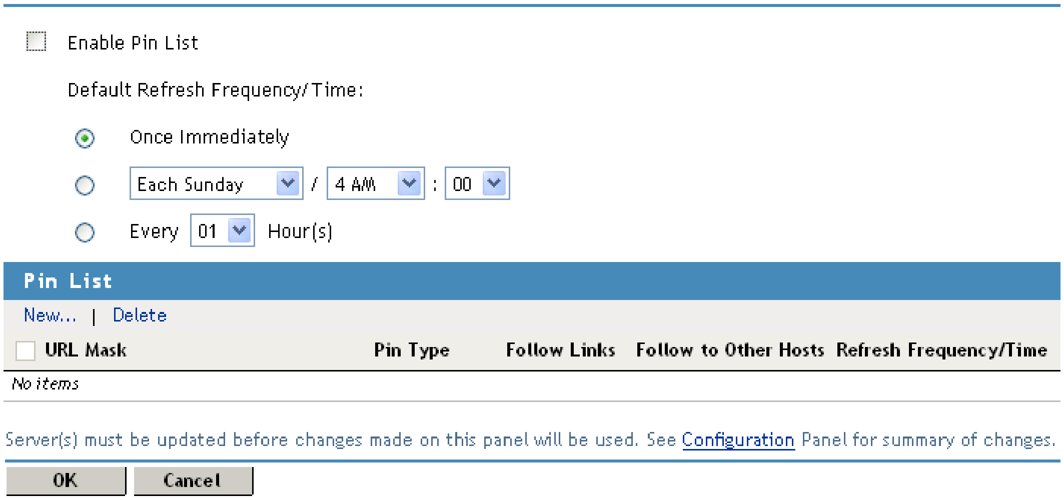
You can use the pin list to specify the following:

- Which objects you want always to remain in cache
- Which objects you never want cached
- (NetWare® only) Whether you want the Access Gateway to follow links on the cached pages and cache these linked objects
- (NetWare only) How often you want the Access Gateway to check for modified content (new and deleted objects)

The pin list is global to the Access Gateway and affects all protected resources. The pinned objects remain in cache indefinitely unless the cache fills up. This ensures that the objects are available from cache and are not bumped out by more recently requested objects. You configure each pinned object with a URL pattern and specific handling instructions.

To configure a pin list:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Pin List*.



☐ Enable Pin List

Default Refresh Frequency/Time:

● Once Immediately

○ Each Sunday / 4 AM : 00

○ Every 01 Hour(s)

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2 Fill in the following fields:

Enable Pin List: Select this option to enable the use of pinned objects. If this option is not selected, the pinned objects in the pin list are not used.

Default Refresh Frequency/Time: (NetWare only) Sets a default refresh interval for checking the URL patterns and seeing if any new objects need to be cached (or deleted objects removed from cache). This default refresh interval can be overwritten by selecting a different refresh interval for a specific pinned object. Select one of the following for the default value:

- **Once Immediately:** Select this option to refresh the list as soon as the changes to this page are pushed to the server.
- **Day and Hour:** Select a day and a time for the refresh.
- **Hourly Interval:** Select an interval, specified in hours, for refreshing the pin list.

3 In the *Pin List* section, click *New*.

4 Fill in the following fields.

URL Mask: Specifies the URL pattern to match. For more information, see Section 16.4.1, “URL Mask,” on page 287.

Pin Type: Specifies how the URL is to be used to cache objects. Select from *Normal*, *Cache*, *Memory*, and *Bypass*. The Linux Access Gateway supports only *Normal* and *Bypass*. For more information, see Section 16.4.2, “Pin Type,” on page 288.

Follow Links: (NetWare only) Indicates whether the Access Gateway can follow links and limits nested links to the value specified. A value of zero indicates that links should not be followed. For more information, see Section 16.4.3, “Follow Links,” on page 289.

Other Hosts: (NetWare only) Indicates whether the Access Gateway can follow links to other hosts and cache pages from these hosts. This is only available if the *Follow Links* field is set between 1 and 4.

Refresh Frequent/Time: (NetWare only) Sets a default refresh interval for checking the URL patterns and seeing if any objects have been modified. You can select *Use Default* to use the refresh interval set for all URL patterns or you can specify one for this object, whose value overrides the default setting.

When the fields are configured, click *OK*.

5 To save your changes to browser cache, click *OK*.

6 To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

16.4.1 URL Mask

The URL mask can contain complete or partial URL patterns. A single URL mask might apply to a large set of URLs, or it might be so specific that only a single file on the Web matches it.

The Access Gateway processes the masks in the pin list in order of specificity. A mask containing a host name is more specific than a mask that specifies only a file type. The action taken for an object is the action specified for the first mask that the object matches.

The Access Gateways recognizes four levels of specificity, using the following format:

Level	**Examples**

hostname	`http://www.foo.gov/documents/picture.gif`
	`http://www.foo.gov/documents/*`
	`http://www.foo.gov`
	`foo.gov/documents/*`
	`foo.gov/*`
	All of these are classified as hostnames, and they are ordered by specificity. The first item in the list is considered the most specific and would be processed first. The last item is the most general and would be processed last.

path	`/documents/picture.gif`
	`/documents/pictures.gif/*`
	`/documents/*`
	Path entries are processed after hostnames. A leading forward slash must always be used when specifying a path, and the entry that follows must always reference the root directory of the Web server. In these examples, `documents` is the root directory.
	The `/*` at the end of the path indicates that the entry is a directory. Its absence indicates that the entry is a file. In these examples, `picture.gif` is a file and `pictures.gif/*` and `documents/*` are directories.
	If you enter a path without the trailing *, the path matches only the directory. With the trailing *, the path matches everything in the directory and its subdirectories.
	These path entry examples are ordered by specificity. The objects in the `/documents/picture.gif` directory are processed before the objects in the `/documents` directory.

filename	`/picture.gif`
	`/widget.js`
	Filenames are processed after paths. A leading forward slash must always be used when specifying a filename. If a path is included with a filename, the path must start with the root directory of the Web server (and the entry is processed as a path entry, not as a filename entry).

Level	Examples

file extension	`/*.gif`
	`/*.js`
	`/*.htm`
	File extensions are processed last. They consist of a leading forward slash, an asterisk, a period, and a file extension.

Specific rules have precedence over less specific rules. Thus, objects matched by a more specific rule are always processed according to its conditions. If a less specific rule also matches the object, the less specific rule is ignored for the object. For example, assume the following two entries in the pin list:

URL Mask	Pin Type	Pin Links
`http://www.foo.gov/documents/*`	cache	1
`www.foo*`	bypass	N/A

The first entry, because it is most specific, caches the pages in the `documents` directory and follows any links on those pages and caches the linked pages. The second entry does not affect what the first entry caches, but it prevents any other domain extensions (.com, .net, .org, etc.) whose DNS names begin with www.foo from being cached.

16.4.2 Pin Type

The pin type specifies how the Access Gateway caches objects that match the URL mask.

- **Normal:** The Access Gateway handles objects matching the mask in the same way it handles any other requested objects. In other words, the objects are cached but not pinned.

  Administrators often use this pin type in combination with a broad URL mask that has a bypass pin type. This allows them to insulate specific objects from the effects of the bypass rule.

  For example, you could specify a URL mask of `/*.jpg` with a pin type of bypass and a second URL mask of `www.foo.gov/graphics/*` with a pin type of normal. This causes all files, including `.jpg` files, in the graphics directory on the `foo.gov` Web site to be cached as requested. They are not, however, pinned in cache because of the normal pin type. Assuming there are no other URL masks in the pin list, all other JPG graphics are not cached because of the `/*.jpg` mask.
- **Cache:** The Access Gateway keeps the pinned objects in cache as long as possible, although they might be written to the hard disk. This option is not supported by the Linux Access Gateway.
- **Memory:** The Access Gateway keeps the pinned objects in memory as long as possible, writes them to disk when memory gets too full, and places them back in memory as soon as they are requested by a user of the cache. This option is not supported by the Linux Access Gateway.
- **Bypass:** The Access Gateway does not cache the objects. In other words, you can use this option to prevent objects from being cached.

16.4.3 Follow Links

The *Follow Links* field specifies the number of links the Access Gateway can follow as it caches objects that match the URL pattern. For example, if the requested object is an HTML page and you have specified a *Follow Links* level of 1, the HTML page is downloaded and cached along with all the items linked from the page. These cached objects are also refreshed at the frequency and time specified. If there are links on the linked pages, these links are not followed and those pages are not cached. To add these objects, you would need to specify 2 for the *Follow Links* option.

To use a level other than 0, you must specify an absolute address, including the scheme, host, and path for the URL mask, for example:

```
http://www.foo.gov/documents/
```

16.5 Configuring a Purge List

The purge list is global to the Access Gateway and affects all protected resources. This option allows you to specify URL patterns or masks for the pages and sites whose objects you want to purge from cache.

When defining the masks, keep in mind that the Access Gateway interprets everything in the URL mask between the asterisk wildcard (*) and the following delimiter as a wildcard. Delimiters include the forward slash (/), the period (.), and the colon (:) characters. For example:

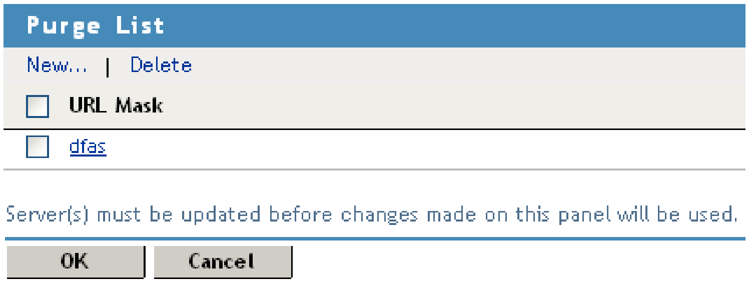
URL Mask	Effects
/*.pdf	Causes all PDF files to be purged from cache.
www.foo.gov/contracts/*	Causes all objects in the `contracts` directory and beyond to be purged from cache.

This option also allows you to purge cached objects whose URL contains a specified query string or cookie. This mask is defined by placing a question mark (?) at the start of the mask followed by text strings and wildcards as necessary. String comparisons are not case sensitive. For example, ?*=SPORTS purges all objects with the text “=SPORTS” or any other combination of uppercase and lowercase letters for “=SPORTS” following the question mark in the URL.

IMPORTANT: If you also configure a pin list, carefully select the objects that you add to the pin and purge lists. You can configure the Access Gateway to use the pin list to add objects to the cache and to use the purge list to remove the same objects.

---

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Purge List*.



Server(s) must be updated before changes made on this panel will be used.

2. Click *New,* enter a URL pattern, then click *OK*.
3. (Optional) Repeat Step 2 to add additional URL patterns.
4. To save your changes to browser cache, click *OK*.
5. To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

16.6 Purging Cached Content

You can select to purge the content of the purge list or all content cached on the server.

1. In the Administration Console, click *Access Manager* > *Access Gateways*.
2. Select the name of the server, then click *Actions*.
3. Select one of the following actions:

   Purge List Now: Click this action to cause all objects in the current purge list to be purged from the cache.

   Purge All Cache: Click this action to purge the server cache. All cached content, including items cached by the pin list, is purged.
4. Click either *OK* or *Cancel*.

When you make certain configuration changes such as updating or changing certificates, changing the IP addresses of Web servers, or modifying the rewriter configuration, you are prompted to purge the cache. The cached objects must be updated for users to see the effects of such configuration changes. If your Access Gateways are in a cluster, you need to manage the purge process so your site remains accessible to your users. You should apply the configuration changes to one member of a cluster. When its status returns to healthy and current, issue the command to purge its cache. Then apply the changes to the next cluster member.

IMPORTANT: Do not issue a purge cache command when an Access Gateway has a pending configuration change. Wait until the configuration change completes.

16.7 Preventing a Web Site from Being Cached

The Access Gateway is designed to cache Web pages. However, sometimes you need to use the Access Gateway to protect a Web site and provide single sign-on, but you do not want the content of the Web server cached.

To prevent the caching of a Web site, you need to add the site to the pin list with a pin type of *Bypass*.

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Pin List*.
2. Make sure the *Enable Pin List* option is selected.
3. In the *Pin List* section, click *New* and fill in the following fields:
   - **URL Mask:** The URL pattern to match. Specify the published DNS name of the Web server that should not have its content cached. For example:

     `http://myserver.mycompany.com`

     This type of entry prevents the caching of pages on the Web site when accessed over HTTP. To block both HTTP and HTTPS, you can add a second entry for HTTPS or remove the scheme from the URL pattern.

- **Pin Type:** The caching action. To prevent caching, select *Bypass*.

For a NetWare Access Gateway, accept the default values for the other fields, or configure them to fit your needs. For more information, see Section 16.4, “Configuring a Pin List,” on page 285.

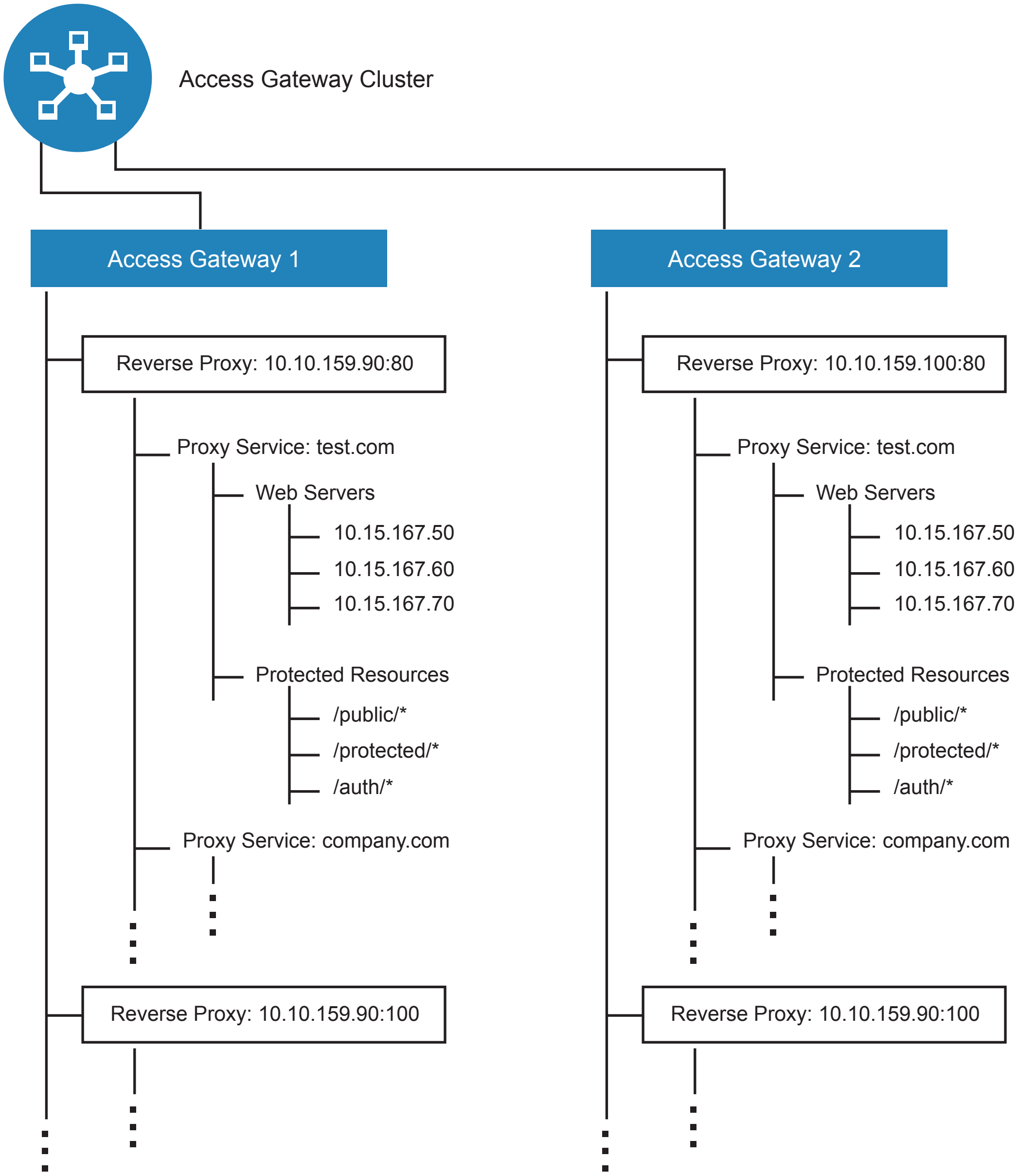
4. Click *OK*.
5. To save your changes to browser cache, click *OK*.
6. To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.
7. To purge any pages that might have been cached while you were configuring the pin list, purge the existing cache. See Section 16.6, “Purging Cached Content,” on page 290.

Protecting Multiple Resources

17

This section describes how to create multiple resources for the various Access Gateway components, including a cluster of Access Gateways. Figure 17-1 illustrates the relationships that Access Gateways, reverse proxies, proxy services, Web servers, and protected resources have with each other when two Access Gateways are members of a cluster.

Figure 17-1 *Hierarchical View of the Access Gateway Configured Objects*



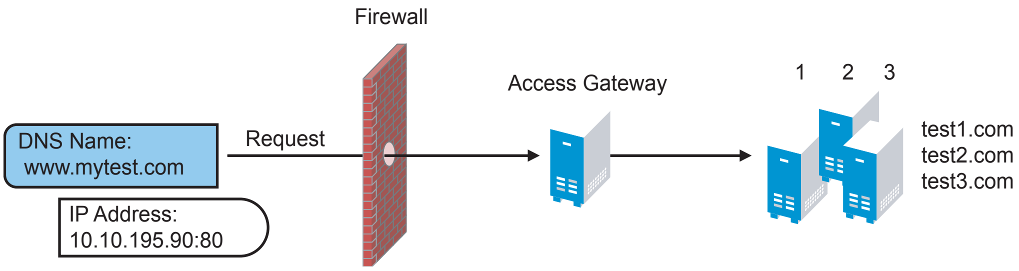
In Figure 17-1, Access Gateway 1 and Access Gateway 2 have the same configuration except for the reverse proxy listening address. They share the other configuration settings because they are members of an Access Gateway cluster. This section explains how to create a group of Web servers, how to add multiple proxy services and reverse proxies to an Access Gateway, and how to manage a cluster of Access Gateways.

- Section 17.1, “Setting Up a Group of Web Servers,” on page 294
- Section 17.2, “Using Multi-Homing to Access Multiple Resources,” on page 295
- Section 17.3, “Managing Multiple Reverse Proxies,” on page 304
- Section 17.4, “Managing a Cluster of Access Gateways,” on page 306

17.1 Setting Up a Group of Web Servers

You can configure a proxy service to service a “virtual” group of Web servers, which adds load balancing and redundancy. Each Web server in the group must contain the same material. When you create the proxy service, you set up the first server by specifying the URLs you want users to access and the rights the users need for each URL. When you add additional Web servers to the proxy service, these servers automatically inherit everything you have configured for the first Web server.

Figure 17-2 *Adding Redundant Web servers*



For this configuration, you use a single reverse proxy and proxy service. To add multiple Web servers to a host:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Web Servers*.
2. In the *Web Server List* section, click *New*.
3. Specify the IP address or the fully qualified DNS name of another Web server for the “virtual” group, then click *OK*.
4. Repeat Steps 2 and 3 to add additional Web servers to the group.
5. To save your changes to browser cache, click *OK*.
6. To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

The NetWare Access Gateway performs a round robin among the Web servers, dividing the load among them. The Linux Access Gateway can perform a round robin, or it can be configured to perform a simple failover, sending all the traffic to the first Web server as long as it is available. Traffic is sent to another Web server in the list only when the first Web server is no longer available. To configure this option, see Section 13.6.2, “Configuring TCP Connect Options for Web Servers,” on page 235.

Connection persistence is enabled by default. This allows the Access Gateway to send multiple HTTP requests to the Web server to be serviced before the connection is closed. To configure this option, see Section 13.6.2, “Configuring TCP Connect Options for Web Servers,” on page 235.

Session persistence is enabled whenever a second Web server is added to the list. This allows a browser to persistently use the same Web server after an initial connection has been established. This type of persistence is not configurable. For more information on persistent connections, see Section 13.6.3, "Configuring Connection and Session Persistence," on page 237.

17.2 Using Multi-Homing to Access Multiple Resources

You can configure the Access Gateway to use one public IP address to protect multiple types of Web resources. This is one of the major benefits of Access Gateway because it conserves valuable resources such as IP addresses. This feature also makes the Access Gateway a multi-homing device because it becomes a single endpoint supporting multiple back-end resources.

You can select to use only one multi-homing method, or you can use multiple methods. Select the methods that meet the needs of your network and the resources you are protecting. The first proxy service configured for a reverse proxy is always configured to use the DNS name of the Access Gateway. Subsequent proxy services can be configured to use one of the following methods:

- Section 17.2.1, "Domain-Based Multi-Homing," on page 295
- Section 17.2.2, "Path-Based Multi-Homing," on page 297
- Section 17.2.3, "Virtual Multi-Homing," on page 299

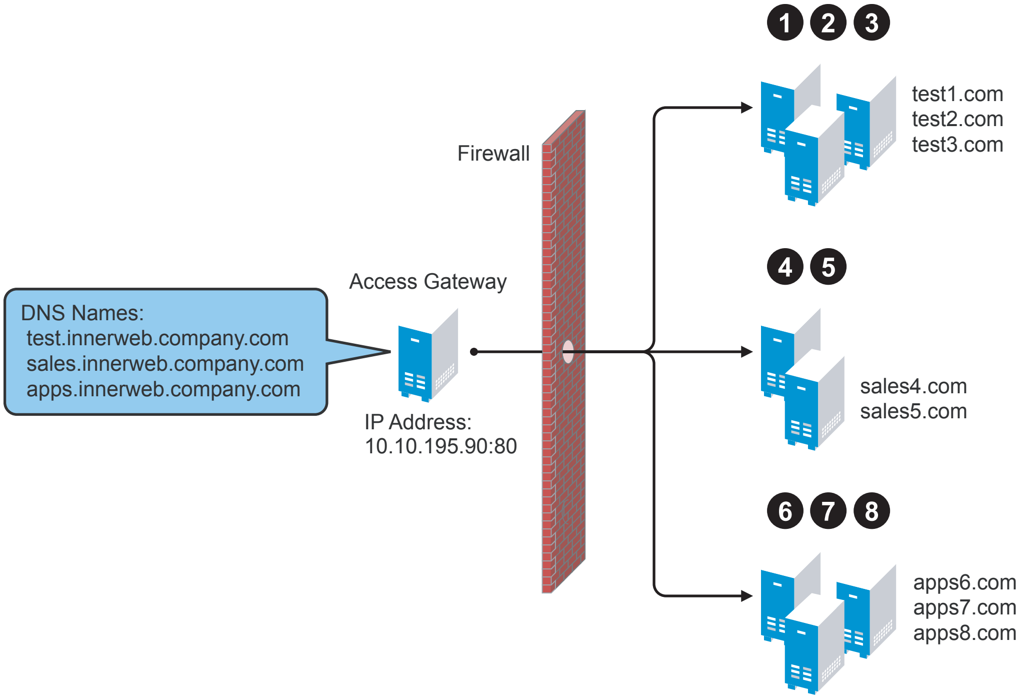
This section describes these multi-homing methods, then explains the following:

- Section 17.2.4, "Creating a Second Proxy Service," on page 300
- Section 17.2.5, "Configuring a Path-Based Multi-Homing Proxy Service," on page 302

17.2.1 Domain-Based Multi-Homing

Domain-based multi-homing is based on the cookie domain. For example, if you have a cookie domain of company.com, you can prepend host names to cookie domain name. For a test resource, you can prepend test to company.com and have test.company.com resolve to the IP address of the Access Gateway. The Access Gateway configuration for the test.company.com proxy service contains the information for accessing its Web servers (test1.com). Figure 17-3 illustrates this type of configuration for three proxy services.

Figure 17-3 *Using a Base Domain Name with Host Names*



Domain-based multi-homing has the following characteristics:

- If you are using SSL, the back-end servers can all listen on the same SSL port (default for HTTPS is 443).
- If you are using SSL, the back-end servers can share the same SSL certificate. Instead of using a specific host name in the SSL certificate, the certificate can use a wildcard name such as *.company.com, which matches all the servers.

Before configuring the Access Gateway, you need to complete the following:

- Create the published DNS names with a common domain name for public access to the back-end resources. For example, the table below lists three DNS that use company.com as a common domain name and then lists the IP address that these DNS names resolve to and the Web servers they are going to protect.

Published DNS Name	**Access Gateway IP Address**	**Web Server Host Name**	**Web Server IP Address**
test.company.com	10.10.195.90:80	test.internal.com	10.15.0.10
sales.company.com	10.10.195.90:80	sales.internal.com	10.15.0.20
apps.company.com	10.10.195.90:80	apps.internal.com	10.15.0.30

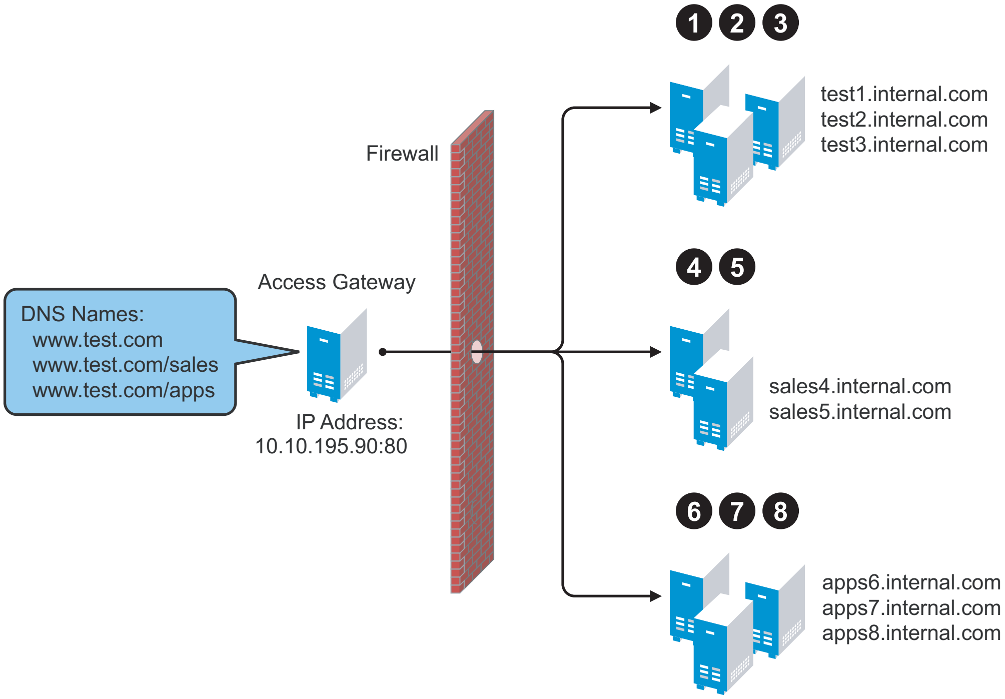
- Configure your DNS server to resolve the published DNS names to the IP address of the Access Gateway.
- Set up the back-end Web servers.

To create a domain-based multi-homing proxy service, see Section 17.2.4, “Creating a Second Proxy Service,” on page 300, and select domain-based for the multi-homing type.

17.2.2 Path-Based Multi-Homing

Path-based multi-homing uses the same DNS name for all resources, but each resource, or resource group, must have a unique path appended to the DNS name. For example, if the DNS name is test.com, you would append /sales to test.com. When the user enters the URL of www.test.com/sales, the Access Gateway resolves the URL to the sales resource group. Figure 17-4 illustrates this type of configuration.

Figure 17-4 *Using a Domain Name with Path Elements*



Path-based multi-homing has the following characteristics:

- It is considered to be more secure than domain-based multi-homing, because some security experts consider wildcard certificates less secure than a certificate with a specific hostname.
- Each resource or group of resources must have a unique starting path.
- JavaScript applications might not work as designed if they obscure the URL path. The Access Gateway needs access to the URL path, and if it is obscured, the path cannot be resolved to the correct back-end resource.
- The protected resources for each path-based child come from the parent proxy service.

The following sections explain how to configure path-based proxy services and your network so that the Access Gateway can find the correct protected resources:

- “Configuring the Remove the Path on Fill Option” on page 298
- “Configuring the Host Header Option” on page 298
- “Configuring for Path-Based Multi-Homing” on page 299

Configuring the Remove the Path on Fill Option

If the path that is part of the published DNS name (/sales or /apps) is used to identify a resource but is not part of directory configuration on the Web server, the path needs to be removed from the URL before the request is sent to the Web server. For example, suppose you use the following configuration:

Browser URL Using the Published DNS Name	Web Server URL
http://www.test.com/sales	http://sales4.internal.com/

In this case, the path needs to be removed from the URL that the Access Gateway sends to the Web server. The Access Gateway does not allow you to set up multiple paths to this type of Web server, so all pages must have the same authentication requirements.

If the path in the published DNS name is a path on the Web server, the path needs to be passed to the Web server as part of the URL. For example, suppose you use the following configuration:

Browser URL Using the Published DNS Name	Web Server URL
http://www.test.com/sales	http://sales4.internal.com/sales

Because the path component specifies a directory on the Web server where the content begins, you need to select to include the path. The Access Gateway then includes the path as part of the URL it sends to the Web server. This configuration allows you to set up multiple paths to the Web server, such as

- sales/payroll
- sales/reports
- sales/products

Such a configuration also allows you to set up different authentication and authorization requirements for each path.

Configuring the Host Header Option

When you create path-based proxy services and also enable the *Remove Path on Fill* option, you need to know what types of links exist on the Web servers. For example, you need to know if the sales Web servers in Figure 17-4 on page 297 have links to the app Web servers or to the test Web servers. If they don't, you can set the *Host Header* option to either *Forward Received Host Name* or to *Web Server Host Name*. However if they do contain links to each other, you need to set the *Host Header* option to *Web Server Host Name* and specify a DNS name for the Web server in the *Web Server Host Name* option. The Access Gateway needs a method to distinguish between the Web servers other than the path, because after the path is removed, all the Web servers in Figure 17-4 on page 297 have the same name: www.test.com.

If you select to use the *Forward Received Host Name* option for a path-based service, you might also need to add entries to the *Additional DNS Name List* for the rewriter. For more information, see "Determining Whether You Need to Specify Additional DNS Names" on page 218.

Configuring for Path-Based Multi-Homing

Before configuring the Access Gateway, you need to complete the following:

- Create the published DNS names with paths for public access to the back-end resources. For example, the table below uses test.com as the domain name. It lists three published DNS names (two with paths), the IP address these names resolve to, and the Web servers that they are going to protect:

Published DNS Name	Access Gateway IP Address	Web Server Host Name	Web Server IP Address
test.com	10.10.195.90:80	test.internal.com	10.15.0.10
test.com/sales	10.10.195.90:80	sales.internal.com	10.15.0.20
test.com/apps	10.10.195.90:80	apps.internal.com	10.15.0.30

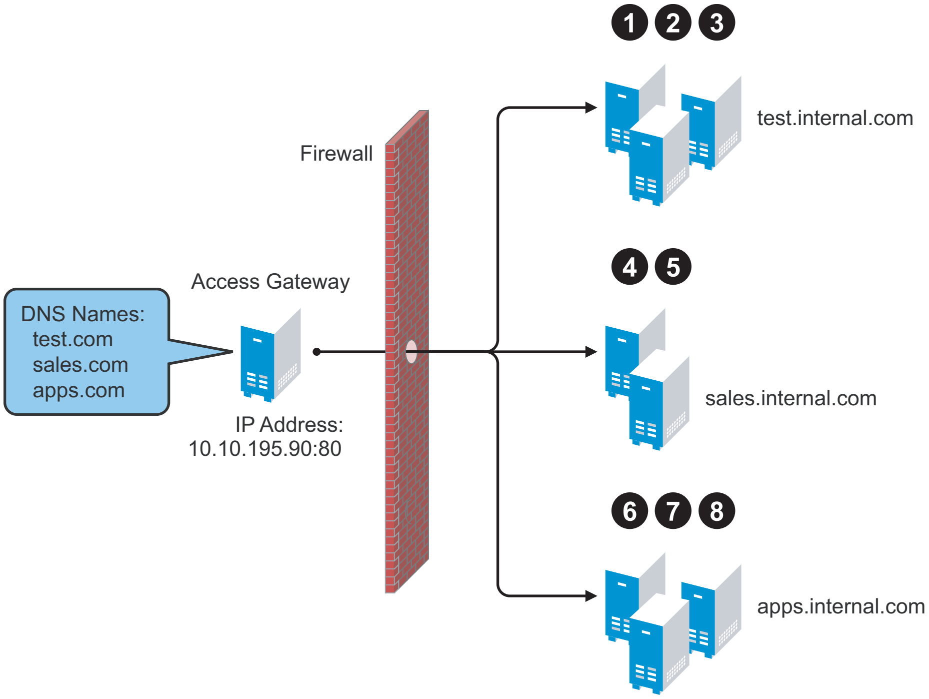
- Configure your DNS server to resolve the published DNS names to the IP address of the Access Gateway.
- Set up the back-end Web servers. If they have links to each other, set up DNS names for the Web servers.

To create a path-based multi-homing proxy service, see Section 17.2.4, “Creating a Second Proxy Service,” on page 300, and select path-based for the multi-homing type.

17.2.3 Virtual Multi-Homing

Virtual multi-homing allows you to use DNS names from different domains (for example test.com and sales.com). Each of these domain names must resolve to the Access Gateway host. Figure 17-5 illustrates this type of configuration.

Figure 17-5 *Using Multiple DNS Names*



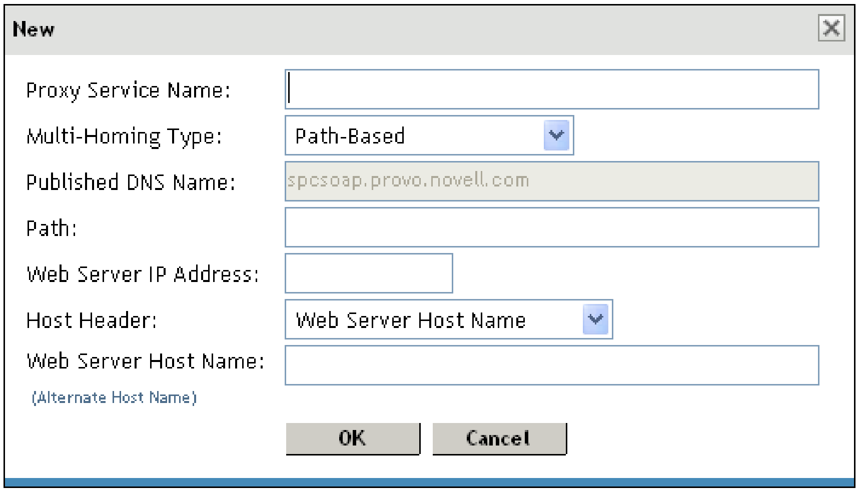
Virtual multi-homing cannot be used with SSL. You should use this configuration with resources that need to be protected, but the information exchanged should be public information that does not need to be secure. For example, you could use this configuration to protect your Web servers that contain the catalog of your shipping products. It isn't until the user selects to order a product that you need to switch the user to a secure site.

Whether a client can use one DNS name or multiple DNS names to access the Access Gateway depends upon the configuration of your DNS server. After you have configured your DNS server to allow this, you are ready to configure the Access Gateway.

To create a Virtual multi-homing proxy service, see Section 17.2.4, "Creating a Second Proxy Service," on page 300, and select *Virtual* for the multi-homing type.

17.2.4 Creating a Second Proxy Service

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.
2. In the *Proxy Service List*, select *New*.



3 Fill in the fields.

Proxy Service Name. Specify a display name for the proxy service. For the sales group, you might use sales. For the group of application servers, you might use apps.

Multi-Homing Type: Specify the multi-homing method that the Access Gateway should use to identify this proxy service. Select one of the following:

- **Domain-Based:** Uses the published DNS name (www.test.com) with a host name (www.newsite.test.com). For more information, see Section 17.2.1, “Domain-Based Multi-Homing,” on page 295.
- **Path-Based:** Uses the published DNS name (www.test.com) with a path (www.test.com/path). For more information, see Section 17.2.2, “Path-Based Multi-Homing,” on page 297.
- **Virtual:** Uses a unique DNS name (www.newsite.newcompany.com). Virtual multi-homing cannot be used with SSL. For more information, see Section 17.2.3, “Virtual Multi-Homing,” on page 299. If you need a unique DNS name and SSL, you need to create a reverse proxy rather than a proxy service. For information on creating a second reverse proxy, see Section 17.3, “Managing Multiple Reverse Proxies,” on page 304.

Published DNS Name: Specify the DNS name you want the public to use to access your site. This DNS name must resolve to the IP address you set up as the listening address. This option is not available when path-based multi-homing is selected.

Path: Specify the path to use for this proxy service. This option is available only when path-based multi-homing is selected.

Web Server IP Address: Specify the IP address of the Web server you want this proxy service to manage.

Host Header: Specify whether the HTTP header should contain the name of the back-end Web server (*Web Server Host Name* option) or whether the HTTP header should contain the published DNS name (the *Forward Received Host Name* option).

For a path-based multi-homing service, it is usually best to select the *Web Server Host Name* option. For more information, see “Configuring the Host Header Option” on page 298.

Web Server Host Name: Specify the DNS name of the Web server that the Access Gateway should forward to the Web server. If you have set up a DNS name for the Web server and the Web server requires its DNS name in the HTTP header, specify that name in this field. If you selected *Forward Received Host Name*, this option is not available.

NOTE: For iChain® administrators, the *Web Server Host Name* is the alternate host name when configuring a Web Server Accelerator.

---

4 Click *OK*.

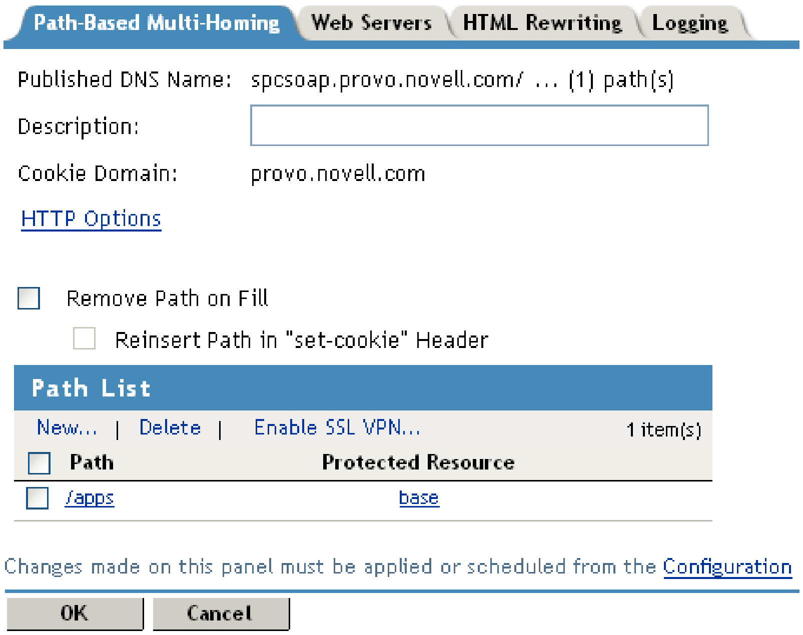
5 To continue, select one of the following:

- To configure a virtual or domain-based proxy service, see Section 13.2, "Configuring a Proxy Service," on page 204.
- To configure a path-based proxy service, see Section 17.2.5, "Configuring a Path-Based Multi-Homing Proxy Service," on page 302.

17.2.5 Configuring a Path-Based Multi-Homing Proxy Service

To configure a path-based proxy service:

1 In the Administration Console, click *Access Manager > Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Path-Based Multi-Homing Proxy Service]*.



Published DNS Name: spcsoap.provo.novell.com/ ... (1) path(s)

Description:

Cookie Domain: provo.novell.com

☐ Remove Path on Fill

☐ Reinsert Path in "set-cookie" Header

Path List

1 item(s)

☐	Path	Protected Resource
☐	/apps	base

Changes made on this panel must be applied or scheduled from the Configuration

The following fields display information that must be configured on the parent proxy service (the first proxy service created for this reverse proxy).

- **Published DNS Name:** Displays the value that users are currently using to access this proxy service. This DNS name must resolve to the IP address you set up as a listening address on the Access Gateway.
- **Cookie Domain:** Displays the domain for which the cookie is valid. The Web server that the user is accessing must be configured to be part of this domain.

2 Configure the following options:

Description: (Optional) Provides a field where you can describe the purpose of this proxy service or specify any other pertinent information.

HTTP Options: Determines how the proxy service handles HTTP headers and caching. For more information, see Section 16.3, "Configuring Custom Cache Control Headers," on page 283, Section 16.2, "Controlling Browser Caching," on page 282, and Section 16.1, "Configuring Global Caching Options," on page 279.

3 Configure the path options:

Remove Path on Fill: Determines whether the multi-homing path is removed from the URL before forwarding it to the Web server. If the path is not a directory at the root of the Web server, the path must be removed. If this option is selected, the path is stripped from the request before the request is sent to the Web server.

If you enable this option, this proxy service can protect only one path. If you have configured multiple paths in the *Path List*, you cannot enable this option until you have deleted all but one path.

Reinsert Path in "set-cookie" Header: Determines whether the path is inserted into the "set cookie" header. This option is only available if you enable the *Remove Path on Fill* option.

4 Determine whether you need to create a protected resource for your path.

In the *Path List*, the path you specified is listed along with the protected resource that best matches its path.

The Access Gateway automatically selects the protected resource that is used with the specified path. It selects the current protected resource whose URL path most closely matches the specified path.

- If you have a protected resource with a URL path of /*, the Access Gateway selects that resource unless you have configured a protected resource that has a URL path that more closely matches the path specified on this page.
- If you add a protected resource at a future time and its URL path more closely matches the path specified on this page, the Access Gateway automatically reconfigures to use this new protected resource.
- If you disable a protected resource that the Access Gateway has assigned to a path-based service, the Access Gateway automatically reconfigures and selects the next protected resource that most closely matches the path specified on this page.

4a In the *Path List* section, click the *Protected Resource* link.

4b Examine the contract, Authorization, Identity Injection, and Form Fill policies assigned to this protected resource.

4c To return to the Path-Based Multi-Homing page, click the *Overview* tab, then click *OK*.

- If the protected resource meets your needs, continue with Step 5
- If it does not meet your needs, you must create a protected resource for the path-based proxy service. Continue with Step 4d.

4d Click *OK*, the name of the parent proxy service, then *Protected Resources*.

4e In the *Protected Resource List*, click *New*, specify a name, then click *OK*.

4f Assign a contract.

4g In the *URL Path List*, specify the path you used when creating the path-based proxy service. For example, if your path was `/apps`, specify `/apps/*` or `/apps` in the URL Path List.

IMPORTANT: If you create multiple protected resources that exactly match the path-based multi-homing service, there is no guarantee which protected resource will be used. For example, if you create protected resources for both of the paths specified above (`/apps` and `/apps/*`) and you have a path-based service with a path of `/apps`, either of these protected resources could be assigned to this path-based service in the Administration Console or used when access is requested.

4h Make sure the protected resource you created is enabled. If the resource is disabled, it does not appear in the Path List for the path-based proxy service.

4i (Optional) Enable the policies the path-based proxy service requires. Click *Authorization*, *Identity Injection*, or *Form Fill* and enable the appropriate policies.

4j Click *OK*.

5 To save your changes to browser cache, click *OK*.

6 To apply the changes, click the *Access Gateways* link, then click *Update > OK*.

17.3 Managing Multiple Reverse Proxies

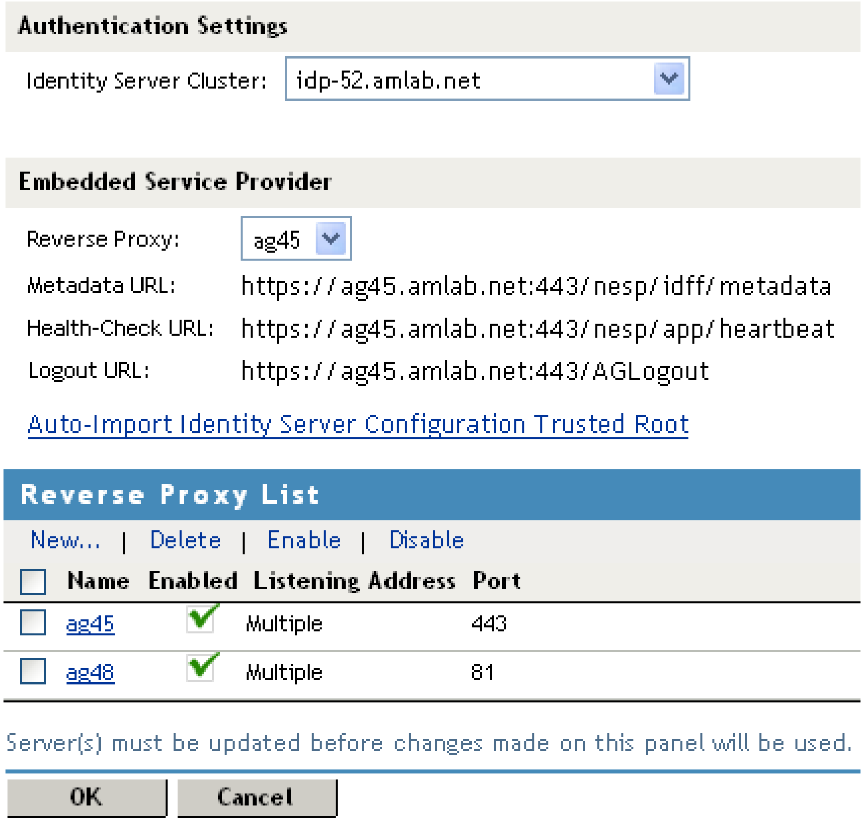
Each reverse proxy must have a unique IP address and port combination. If your Access Gateway has only one IP address, you must select unique port numbers for each additional reverse proxy that you create. You can configure the Access Gateway to use multiple IP addresses. These addresses can be configured to use the same network interface card, or if you have installed multiple network cards, you can assign the IP addresses to different cards. To configure IP addresses and network interface cards, see Section 15.7.1, “Viewing and Modifying Adapter Settings,” on page 264.

If you are creating more than one reverse proxy, you must select one to be used for authentication. By default, the first reverse proxy you create is assigned this task. Depending upon your Access Gateway configuration, you might want to set up one reverse proxy specifically for handling authentication. The authentication reverse proxy is also used for logout. If you have Web applications that contain logout options, these options need to be redirected to the Logout URL of the authentication proxy.

- Section 17.3.1, “Managing Entries in the Reverse Proxy List,” on page 304
- Section 17.3.2, “Changing the Authentication Proxy Service,” on page 306

17.3.1 Managing Entries in the Reverse Proxy List

1 In the Administration Console, click *Access Manager > Access Gateways > Edit > Reverse Proxy / Authentication*



Authentication Settings

Identity Server Cluster: idp-52.amlab.net

Embedded Service Provider

Reverse Proxy: ag45

Metadata URL: https://ag45.amlab.net:443/nesp/idff/metadata

Health-Check URL: https://ag45.amlab.net:443/nesp/app/heartbeat

Logout URL: https://ag45.amlab.net:443/AGLogout

Auto-Import Identity Server Configuration Trusted Root

Reverse Proxy List

New... | Delete | Enable | Disable

☐	Name	Enabled	Listening Address	Port
☐	ag45	✓	Multiple	443
☐	ag48	✓	Multiple	81

Server(s) must be updated before changes made on this panel will be used.

2 In the *Reverse Proxy List*, select one of the following actions:

- **New:** To create a new reverse proxy, click *New*. You are prompted to enter a display name for the proxy. For configuration information, see Section 13.1, “Creating a Reverse Proxy and Proxy Service,” on page 200.

  Reverse proxy names and proxy service names must be unique to the Access Gateway. Protected resource names need to be unique to the proxy service, but they don’t need to be unique to the Access Gateway.
- **Delete:** To delete a reverse proxy, select the check box by a specific reverse proxy, then click *Delete*. To delete all reverse proxies, select the check box by the *Name* column, then click *Delete*.
- **Enable:** To enable a reverse proxy, select the check box by a specific reverse proxy, then click *Enable*. To enable all reverse proxies, select the check box by the *Name* column, then click *Enable*.
- **Disable:** To disable a reverse proxy, select the check box by a specific reverse proxy, then click *Disable*. To enable all reverse proxies, select the check box by the *Name* column, then click *Disable*.

3 To save your changes to browser cache, click *OK*.

4 To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.

17.3.2 Changing the Authentication Proxy Service

If you have multiple reverse proxies, you can select the reverse proxy that users are redirected to for login and logout.

IMPORTANT: Changing the reverse proxy that is used for authentication is not a trivial task. For example, if you have customized the logout options on your Web servers to redirect the logout request to the Logout URL of the current authentication reverse proxy, you need to modify these options to point to a new Logout URL.

If you have set up SSL connections, you need to change your certificate configurations.

To select the reverse proxy to use for authentication:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Reverse Proxy / Authentication*.
2. In the *Embedded Service Provider* section, select a value for the *Reverse Proxy* option. This is the reverse proxy that is used for authentication.

   The screen is refreshed and the *Metadata URL*, *Health-Check URL,* and *Logout URL* are rewritten to use the selected reverse proxy.
3. (Conditional) If your Access Gateway certificates were generated by a different certificate authority than your Identity Server certificates, you need to import the trusted root of the Identity Sever into the trusted root keystore of the embedded service provider. Click *Auto-Import Identity Server Configuration Trusted Root*, click *OK*, specify an alias, click *OK*, then click *Close*.

   If you don't know whether you need to import the trusted root, click the option. If the trusted root is already in the keystore, the duplicate key is not imported and you are informed of this condition.
4. In the *Reverse Proxy List*, click the name of the reverse proxy that you have selected for authentication.
5. If you have enabled SSL between the embedded service provider and the Identity Server, you need to import the trusted root of the embedded service provider into the trusted root keystore of the Identity Server. Click *Auto-Import Embedded Service Provider Trusted Root*, click *OK*, specify an alias, click *OK*, then click *Close*.

   If you don't know whether you need to import the trusted root, click the option. If the trusted root is already in the keystore, the duplicate key is not imported and you are informed of this condition.
6. To save your changes to browser cache, click *OK*.
7. To apply the changes, click the *Access Gateways* link, then click *Update* > *OK*.
8. (Conditional) If you have customized Web logout pages, update them to use the new Logout URL.

17.4 Managing a Cluster of Access Gateways

Most of the configuration tasks are the same for a single Access Gateway and a cluster of Access Gateways. (For information on how to create a cluster of Access Gateways, see "Clustering Access

Gateways" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.) This section describes the tasks that are specific to managing the servers of an existing cluster:

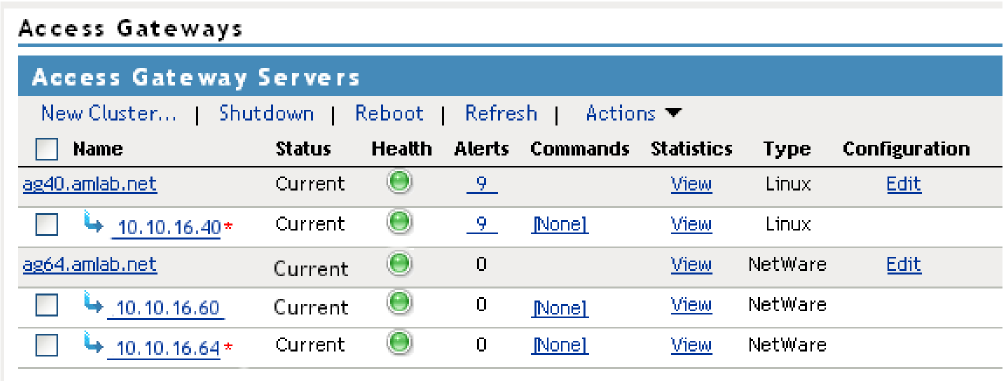
- Section 17.4.1, "Managing the Servers in the Cluster," on page 307
- Section 17.4.2, "Changing the Primary Cluster Server," on page 308
- Section 17.4.3, "Applying Changes to Cluster Members," on page 308

For information about monitoring the health or statistics of a cluster, see Part VII, "Monitoring Access Manager Components," on page 503.

17.4.1 Managing the Servers in the Cluster

To view the servers that are currently members of clusters:

1 In the Administration Console, click *Access Manager* > *Access Gateways*.



☐ Name	Status	Health	Alerts	Commands	Statistics	Type	Configuration
ag40.amlab.net	Current	●	9		View	Linux	Edit
☐ ↳ 10.10.16.40*	Current	●	9	[None]	View	Linux	
ag64.amlab.net	Current	●	0		View	NetWare	Edit
☐ ↳ 10.10.16.60	Current	●	0	[None]	View	NetWare	
☐ ↳ 10.10.16.64*	Current	●	0	[None]	View	NetWare	

The members of a cluster are listed under the cluster name. The asterisk marks the server that is the primary cluster server.

2 To add a server to a cluster, select the server, then click *Actions* > *Assign to Cluster* > *[Name of Cluster]*.

3 To remove a server from a cluster, select the server, then click *Actions* > *Remove from Cluster*.

Usually when you delete a server from a cluster, you have discovered that traffic is lighter than anticipated and that it can be handled with fewer machines while another cluster is experiencing higher traffic and can benefit from having another cluster member. When the server is removed, its configuration object maintains all the configuration settings from the cluster. When it is added to a new cluster, its configuration object is updated with the configuration settings of the new cluster. If your clusters are behind an L4 switch, you need to reconfigure the switch so that the server is assigned to the correct cluster.

When a server is removed from a cluster, its embedded service provider is stopped. If you are not going to assign it to another cluster, you need to reconfigured the server so that it is protecting resources other than the ones it did in the cluster. When you apply the changes by clicking *Update*, the embedded service provider is restarted.

4 To modify which server is the primary cluster server, see Section 17.4.2, "Changing the Primary Cluster Server," on page 308.

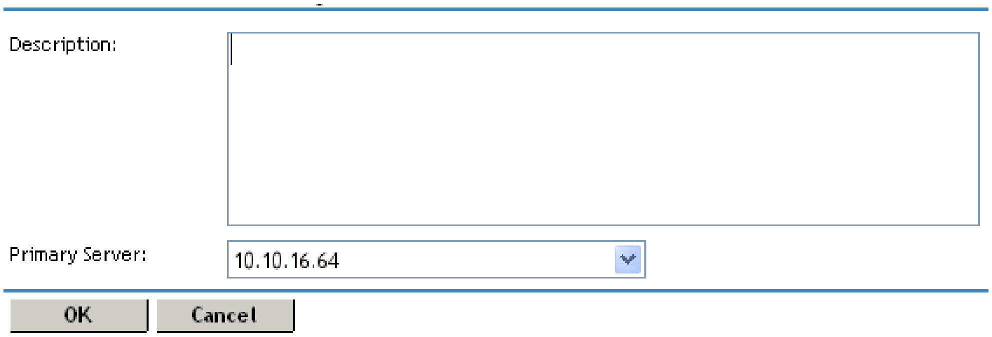
5 To view detailed information about a server in the group, click the name of the server.

6. To view detailed health information about a server, click the health icon of the server. For more information, see Section 34.3, "Monitoring the Health of an Access Gateway," on page 553.
7. Click *Close*.

17.4.2 Changing the Primary Cluster Server

If the current primary cluster server is down and will be down for an extended period of time, you should select another server to be the primary cluster server

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Cluster]* > *Edit*.



2. In the *Primary Server* drop-down list, select the name of a server, then click *OK*.
3. To update the Identity Server, click *Identity Servers* > *Update*.

17.4.3 Applying Changes to Cluster Members

When you are configuring services of the Access Gateway, the *OK* button saves the change to browser cache except on the Configuration page. The Configuration page (*Access Manager* > *Access Gateways* > *Edit*) provides a summary of the changes you have made. The *Cancel Change* column allows you to cancel changes to individual services. When you click *OK*, the changes are saved to the configuration datastore and you no longer have the option to cancel changes to individual services.

When servers are in a cluster, you might want to update only one server in the cluster and verify that the changes are behaving as expected. If this is your plan, we highly recommend that you save the proposed changes to the configuration datastore so the changes are not lost. If your session times out or you log out, any configuration changes that are saved to browser cache are flushed. These changes cannot be applied to other members of the cluster because they are no longer available. To prevent this from happening, save the changes to the configuration datastore.

After testing the configuration on one server, you can then apply the saved changes to the other servers in the cluster, either individually (with the *Update* link) or as group (with the *Update All* link).

If you discover that the configuration change is not behaving the way you want it to, you can revert back to the previous applied configuration by doing the following:

1. Remove the server that you have applied the configuration changes from the cluster.
2. Access the Configuration page for the cluster, then click *Revert*.

The servers in the cluster revert to the last applied configuration.

3 Add the removed server to the cluster.

The server is configured to use the same configuration as the other cluster members.

When you make the following configuration changes, the *Update All* option is the only option available and your site is unavailable while the update occurs:

- The Identity Server configuration that is used for authentication is changed (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication,* then select a different value for the *Identity Server Cluster* option).
- A different reverse proxy is selected to be used for authentication (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication*, then select a different value for the *Reverse Proxy* option).
- The protocol or port of the authenticating reverse proxy is modified (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication* > *[Name of Reverse Proxy]*, then change the SSL options or the port options).
- The published DNS name of the authentication proxy service is modified (*Access Gateways* > *Edit* > *Reverse Proxy/Authentication* > *[Name of Reverse Proxy]* > *[Name of First Proxy Service]*, then modify the *Published DNS Name* option).

SSL VPN Gateway Configuration

IV

The Novell® Access Manager SSL VPN is a Linux-based Virtual Private Network (VPN) service. It provides authorized and secure access to HTTP and non-HTTP-based applications and services that are behind the firewall. This solution uses a Web browser as the primary client interface.

Before you proceed with the SSL VPN configuration, you must have done the following:

- Installed the SSL VPN server. For more information on installing SSL VPN, see the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
- Installed the Access Gateway. For more information on installing Access Gateway, see the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
- Logged in to the Administration Console as the admin user. For more information, see "Logging In to the Administration Console" in *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
- Created an Identity Server configuration. For more information, see Chapter 6, "Configuring an Identity Server," on page 55.

The following sections describe how you can configure and manage the Novell SSL VPN Gateway:

- Chapter 18, "Overview of SSL VPN Services," on page 313
- Chapter 19, "Configuring Basic Setup," on page 321
- Chapter 20, "Controlling Access," on page 329
- Chapter 21, "Managing Server Settings," on page 335
- Chapter 22, "Configuring SSL VPN for Citrix Clients," on page 349

Overview of SSL VPN Services

18

The Novell® Access Manager SSL VPN uses Secure Sockets Layer (SSL) as the underlying security protocol for network transmissions. It uses encryption and other security mechanisms to ensure that the data cannot be intercepted and only authorized users have access to the network.

The Novell SSL VPN server is accelerated by a reverse proxy, which can either be a NetWare® Access Gateway or a Linux Access Gateway.

The Novell SSL VPN uses both clientless and thin-client access methods to ensure that the administrator does not need to do special configuration for users to gain access to applications. Users can access SSL VPN services from any Web browser.

- Section 18.1, "Server Module," on page 313
- Section 18.2, "Client Modes," on page 317
- Section 18.3, "High Bandwidth Version," on page 320

18.1 Server Module

The SSL VPN server is made up of a servlet and a server module. By default, the server module is installed on the machine in which the SSL VPN servlet is running, but you can install the server module on a different machine. The servlet communicates with the SSL VPN server over TCP™ port 2010.

When users access SSL VPN through the Web browser, they are prompted to authenticate. The identity information provided by the user is exchanged between the Access Gateway and the SSL VPN server. On successful authentication, a Java agent or an ActiveX agent is delivered to the client, depending on the browser. This agent establishes a secure tunnel between the user's machine and the SSL VPN server.

This section has the following information:

- Section 18.1.1, "SSL VPN Modes," on page 313
- Section 18.1.2, "How SSL VPN Protects Resources," on page 314
- Section 18.1.3, "Configuring the SSL VPN Servers," on page 316

18.1.1 SSL VPN Modes

The Novell Access Manager SSL VPN is available in two modes, namely, the Enterprise mode and the Kiosk mode. The two modes are available depending on whether you have the administrator right in a Windows workstation or a `root` user privilege on Linux or Macintosh workstations, or if you are a user without administrator rights or `root` user privileges.

This section has the following information:

- "Kiosk Mode" on page 314
- "Enterprise Mode" on page 314

Kiosk Mode

In the Kiosk mode of SSL VPN, only a limited set of applications are enabled for SSL VPN. A non-admin or a non-`root` user who does not have the administrator access can connect to SSL VPN in the Kiosk mode. In Kiosk mode, applications that were opened before the SSL VPN connection was established are not SSL-enabled.

The Kiosk mode supports TCP and UDP applications only. This mode is better suited for machines that are not managed by an organization, such as home computers and computers in Web browsing kiosks.

Enterprise Mode

You can access SSL VPN in the Enterprise mode if you have admin or `root` user access to the workstation, if you know the admin or `root` user credentials, or if you have preinstalled the client components on the workstation.

In Enterprise mode, all applications, including those on the desktop and the toolbar are SSL-enabled, regardless of whether they were opened before or after connecting to SSL VPN. In this approach, a thin client is installed on your workstation. This thin client takes care of the administrator activities required for the Enterprise mode of SSL VPN. In the Enterprise mode, the IP Forwarding feature is enabled by default.

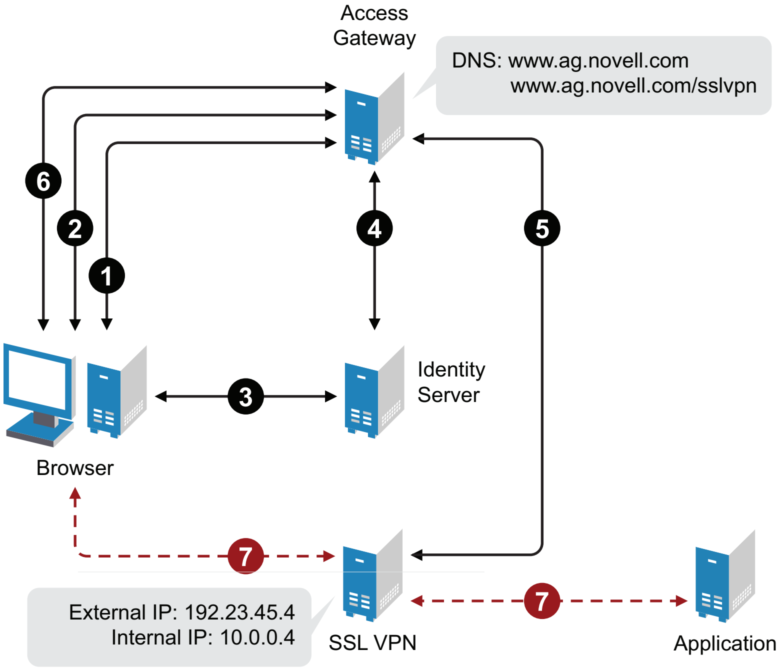
The Enterprise mode is recommended for devices that are managed by an organization, such as a laptop provided by the organization for its employees. The Enterprise mode of SSL VPN supports the following:

- Protocols such as TCP, UDP, ICMP, and NetBIOS.
- Applications that open TCP connections on both sides, such as VoIP and FTP.
- Enterprise applications such as CRM and SAP*.
- Applications such as Windows File Sharing systems, the Novell Client™ and Novell SecureLogin.

18.1.2 How SSL VPN Protects Resources

The following figure shows the Novell Access Manager components and the process involved in establishing a secure connection between a client machine and an SSL VPN server:

Figure 18-1 *How SSL VPN Functions*



1. The user specifies the following URL to access the SSL VPN server:

   https://<*www.ag.novell.com*>/sslvpn/login

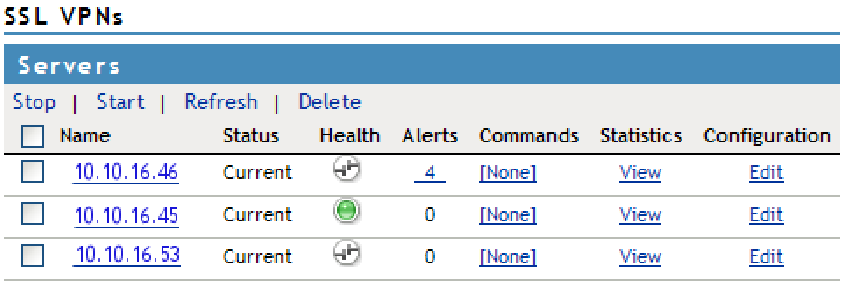
   Here, <*www.ag.novell.com*> indicates the DNS name of the Access Gateway that accelerates the SSL VPN server, and /sslvpn/login indicates the path of the SSL VPN server.
2. The Access Gateway redirects the user to the Identity Server for authentication, because the URL is configured as a protected resource.
3. The Identity Server authenticates the user's identity.
4. The Identity Server propagates the session information to the Access Gateway through the Embedded Service Provider.
5. The Access Gateway injects the SSL VPN policy for that user into the SSL VPN servlet. The SSL VPN servlet processes the parameters and sends the policy information back to the Access Gateway.
6. The SSL VPN checks if the client machine has sufficient security restraints. For more information on client integrity checks, see Chapter 20.2, "Configuring Client Integrity Check Policy to Protect the Internal Network," on page 331.
   a. In Enterprise mode, a tunnel interface is created and is bound with the tunnel IP address assigned by the SSL VPN server. A secure tunnel is established between the client machine and the SSL VPN server and the routing table is updated with the protected network configuration.
   b. In Kiosk mode, a secure tunnel is established between the client machine and the SSL VPN server and the protected network configuration is pushed to the client.
7. When the user accesses the applications behind the protected network, the connection goes through the secure tunnel formed with the SSL VPN server and not through the Access Gateway.

8. Keep the browser open throughout the SSL VPN connection to allow the keep alive packets to go through the Access Gateway.
9. When the user clicks the logout button to close the SSL VPN session, all the client components are automatically uninstalled from the workstation.

18.1.3 Configuring the SSL VPN Servers

SSL VPN servers are auto-imported into the Administration Console during installation.You can use the SSL VPNs page in the Administration Console to view information about the current status of all SSL VPN servers and to configure the SSL VPN servers.

When you click the *SSL VPNs* link in the Administration Console, the following page appears:



SSL VPNs

Servers

Stop | Start | Refresh | Delete

☐	Name	Status	Health	Alerts	Commands	Statistics	Configuration
☐	10.10.16.46	Current	⊕	4	[None]	View	Edit
☐	10.10.16.45	Current	●	0	[None]	View	Edit
☐	10.10.16.53	Current	⊕	0	[None]	View	Edit

The following server information is displayed:

- **Name:** Displays a list of servers added to Administration Console. Click the particular server to view or modify its configuration. For more information, see Section 21.5, “Modifying SSL VPN Server Details,” on page 345.
- **Status:** Indicates the configuration status of the SSL VPN server. Possible states are pending, update, and current. Current indicates that all configuration changes have been applied. Update indicates that a configuration change has been made, but not applied. Click this link to apply the changes. Depending upon what has been modified, updating the complete configuration might cause logged-in users to lose data and lose their connection. Pending indicates that the server is processing a configuration change, but has not completed the process.
- **Health:** Indicates whether the server is functional. Click the icon to view additional information about the operational status of the server. For more information, see Section 34.5, “Monitoring the Health of an SSL VPN Server,” on page 557.
- **Alerts:** Indicates if any alert was sent. This option is not available to you if the alert count is 0. For more information, see Section 36.3, “Monitoring SSL VPN Alerts,” on page 574.
- **Commands:** Indicates the status of commands issued to all servers. For more information, see Section 35.3, “Viewing Command Status of the SSL VPN Server,” on page 561.
- **Statistics:** Indicates the number of active client connections and the time when the server was started. Click *View* to get the statistics information. For more information see, Section 33.3, “Viewing SSL VPN Statistics,” on page 547.
- **Configuration:** Click *Edit* in the *Configuration* column of the SSL VPNs page to view and modify the configuration of the SSL VPN server. This page specifies the date and time when the last modification was made and lists the full distinguished name of the user who made the last modification. For more information, see Chapter 19, “Configuring Basic Setup,” on page 321.

18.2 Client Modes

The client can access SSL VPN either in Enterprise mode or in Kiosk mode, depending on whether users have the administrator right in a Windows workstation or a `root` user permission in Linux or Macintosh workstations, or if they are users without administrator rights or `root` privileges.

This section has the following information:

- Section 18.2.1, "Enterprise Mode," on page 317
- Section 18.2.2, "Kiosk Mode," on page 318
- Section 18.2.3, "User Account Control Feature of Windows Vista and SSL VPN Connection," on page 319

18.2.1 Enterprise Mode

Clients can access SSL VPN in the Enterprise mode in the following scenarios:

- "Scenario 1: Admin or Root User" on page 317
- "Scenario 2: Non-admin or Non-Root User Who Has Admin or Root Credentials" on page 317
- "Scenario 3: Non-Admin or Non-Root User Who Has Preinstalled the Client Components" on page 318

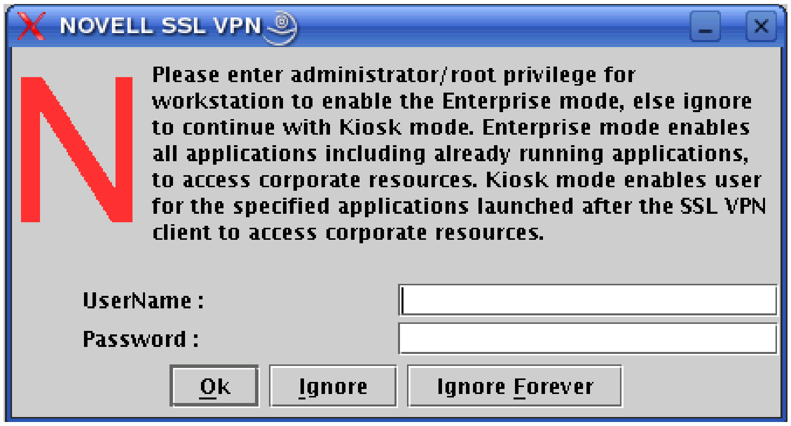
Scenario 1: Admin or Root User

When you are an administrator or a `root` user of the machine, the tool identifies you as the admin or `root` user and the Enterprise mode of SSL VPN is enabled by default. You are connected to the SSL VPN in the Enterprise mode after you specify your credentials in the Access Manager page. An admin or a `root` user can connect to SSL VPN only in the Enterprise mode unless the system administrator configures the users to connect only in Kiosk mode. For more information on how to configure only Kiosk mode to users, see Section 21.1.2, "Configuring SSL VPN to Connect Only in Kiosk Mode," on page 336.

Scenario 2: Non-admin or Non-Root User Who Has Admin or Root Credentials

A non-admin or a non-`root` user can access SSL VPN in the Enterprise mode if the user knows the administrator or `root` user credentials. When a non-admin or a non-`root` user connects to SSL VPN, the user is prompted to specify the credentials on the Access Manager page. The tool identifies that the credentials supplied are those of the non-admin or a non-`root` user and displays the following dialog box.

Figure 18-2 *SSL VPN dialog box*



Please enter administrator/root privilege for workstation to enable the Enterprise mode, else ignore to continue with Kiosk mode. Enterprise mode enables all applications including already running applications, to access corporate resources. Kiosk mode enables user for the specified applications launched after the SSL VPN client to access corporate resources.

UserName :

Password :

The user must specify the username and password of the administrator or the `root` user of the workstation in the dialog box, then click *OK* to enable the Enterprise mode.

The Enterprise mode is enabled by default in the subsequent sessions and the user is not prompted again for the administrator or `root` username and password.

If non-admin or non-`root` users who have connected to SSL VPN in the Enterprise mode want to connect to SSL VPN in Kiosk mode on the same machine, they must follow a certain procedure to do so. For more information, see "Switching from Enterprise Mode to Kiosk Mode" in the *Novell Access Manager 3.0 SP 3 SSL VPN User Guide*.

Scenario 3: Non-Admin or Non-Root User Who Has Preinstalled the Client Components

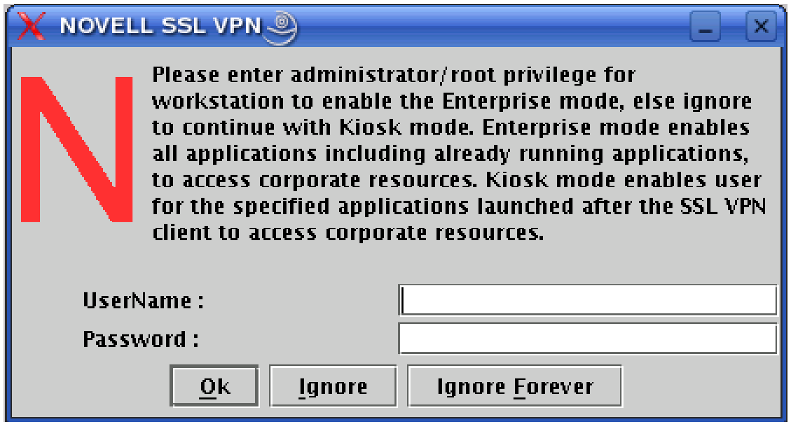
If a non-admin or a non-`root` user wants to install SSL VPN in Enterprise mode, you can preinstall the SSL VPN client components on the user's machine. For more information, see "Pre-Installing SSL VPN Client Components" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*. When non-admin or non-`root` users access the client components from a workstation that has the SSL VPN client components preinstalled, the users are not prompted to enter the credentials of the admin user or `root` user.

The users are connected to SSL VPN in the Enterprise mode after they specify their credentials on the Access Manager login page.

18.2.2 Kiosk Mode

When a user logs in to the SSL VPN client as a non-admin or non-`root` user, the following dialog box is displayed:

Figure 18-3 SSL VPN dialog box



Please enter administrator/root privilege for workstation to enable the Enterprise mode, else ignore to continue with Kiosk mode. Enterprise mode enables all applications including already running applications, to access corporate resources. Kiosk mode enables user for the specified applications launched after the SSL VPN client to access corporate resources.

UserName :

Password :

The user can do one of the following to load the Kiosk mode of SSL VPN:

- Click *Ignore* to connect to SSL VPN in Kiosk mode for that particular session. The user is prompted again to provide the administrator or the root username and password during the next login.
- Click *Ignore Forever* to connect to SSL VPN in Kiosk mode in the current session, as well as in the subsequent sessions.

When the user has clicked *Ignore Forever* and want to connect to SSL VPN in Enterprise mode in the next session, the user has to follow a special procedure. For more information, see “Switching from Kiosk Mode to Enterprise Mode” in *Novell Access Manager 3.0 SP 3 SSL VPN User Guide*.

NOTE: When a non-admin user uses the Internet Explorer to establish SSL VPN connection for the first time, the ActiveX download fails. This happens because the ActiveX requires admin rights to download Activex. This issue might also occur if you have upgraded from an older version. If want to access SSL VPN by using the Internet Explorer, use the following URL:

https:<*DNS-Name*>/sslvpn/login?forcejre

For more information, see Section 21.1.1, “Configuring SSL VPN to Download the Applet on Internet Explorer,” on page 335.

18.2.3 User Account Control Feature of Windows Vista and SSL VPN Connection

The UAC feature is enabled by default on Windows Vista.*. If you are a Windows Vista user and want to access SSL VPN, one of the following scenarios occur depending on the type of the user category that you belong to:

Super User: This is the first user account that is created when Windows Vista is installed on the system. This is the Administrator account and has the right to install or un-install one or more programs, new hardware, and drivers. If a user is a Super User of the machine, then SSL VPN connection is established in the Enterprise mode. If SSL VPN is configured to connect only in the Kiosk mode, the connection is made in the Kiosk mode.

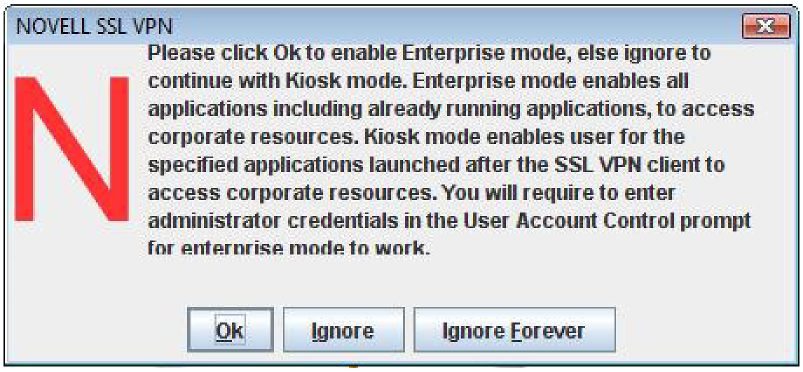
Administrator Category User: An Administrator Category User has the rights do what the Super User does. But, when UAC is enabled in Vista, the users created under Administrator Category are prompted to confirm any changes made to the system settings or configuration or during installation or uninstallation of any software or hardware.

When an Administrator Category User makes an attempt to establish the SSL VPN connection, the user is prompted to confirm if the installation of a certain service can be made on the machine. The user can click Allow or Continue, depending on the prompt, to continue with the SSL VPN installation. If the user allows installation of all the SSL VPN components, the SSL VPN connection is established in the Enterprise mode. If SSL VPN is configured to connect only in the Kiosk mode, the connection is made in the Kiosk mode.

Standard Users: A user created under the Standard User category has minimal privileges. When UAC is enabled, the user is prompted for the Administrator password for any changes that the user intends to make to the system settings.

When a Standard user attempts to connect to SSL VPN, the user is prompted with the following dialog box:

Figure 18-4 *SSL VPN Connection Prompt Dialog Box for Standard User*



Please click Ok to enable Enterprise mode, else ignore to continue with Kiosk mode. Enterprise mode enables all applications including already running applications, to access corporate resources. Kiosk mode enables user for the specified applications launched after the SSL VPN client to access corporate resources. You will require to enter administrator credentials in the User Account Control prompt for enterprise mode to work.

If the user clicks *OK*, the user is prompted to provide the Administrator credentials. If the credentials are valid, then the user is connected to SSL VPN in Enterprise mode. The user is connected to SSL VPN in Kiosk mode if the user clicks Ignore or Ignore Forever in this dialog box.

18.3 High Bandwidth Version

The default SSL VPN server is restricted to 249 simultaneous user connections and a transfer rate of 44 Mbits per second. If the export law permits, you can install the high bandwidth version of SSL VPN, as that version does not have the connection and performance restrictions. You can order the high bandwidth version of SSL VPN at no extra cost.

Configuring Basic Setup

19

The Gateway Configuration page displays the current configuration of the SSL VPN server, such as the external IP address if the SSL VPN server is behind NAT, the listening IP address, TCP encryption port, connection manager port, and the type of encryption used.

19.1 Configuring the Default Identity Injection Policy

The SSL VPN server requires a user credential profile consisting of the following elements:

- Username and password information
- A proxy session cookie
- The roles assigned to the current user for authentication information

Each element added to the custom header requires a name with an "X-" prefix. The name you enter is specific to the application using the custom header, and might be case sensitive. You need to obtain this information from the application before creating the custom header. The Access Gateway injects these headers into the SSL VPN server.

The SSL VPN server requires the following three headers:

- Authentication header containing the credential profile with a username and password
- Custom header containing a proxy session cookie element named X-SSLVPN-PROXY-SESSION-COOKIE
- Custom header containing roles for current user element, named X-SSLVPN-ROLE

You can configure Access Gateway to inject the client IP address as a custom header along with the other three headers. This custom header should be named X-SSLVPN-CLIENTIP. This enables logging of the client IP address for SSL VPN.

NOTE: This is an optional configuration and is not enabled by default. If it is not enabled, the SSL VPN server reports it to the Audit server as a connection accepted from Unknown Host.

To add this header to the SSL VPN policy:

1. In the Administration Console, click *Access Managers* > *Policies*.
2. (Conditional) If you have not created the SSL VPN default policy, click *Create SSL VPN Default*. Then click *Apply Changes*.
3. In the list of policies, click *SSLVPN Default* > 1.
4. In the *Actions* section, click *New*, then select *Inject into Custom Header*.

5 Fill in the following values:

Custom Header Name: Specify *X-SSLVPN-CLIENTIP*.

Value: Select *Client IP*.

6 Click *OK* twice.

7 Click *Apply Changes*.

19.2 Configuring the IP Address, Port, and NAT

This section describes how to configure the IP addresses, port, subnet address and subnet mask, and protocol for SSL VPN.

- Section 19.2.1, "Configuring the SSL VPN Gateway without NAT," on page 322
- Section 19.2.2, "Configuring the SSL VPN Gateway behind NAT," on page 324

19.2.1 Configuring the SSL VPN Gateway without NAT

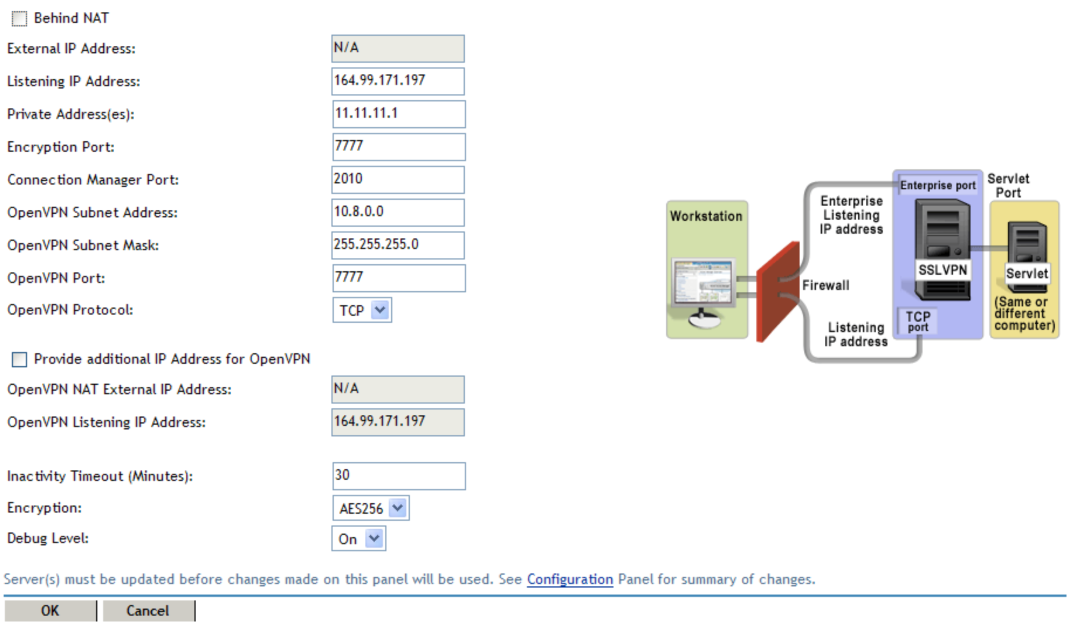
If your SSL VPN gateway is not configured behind NAT (Network Address Translation), complete the following procedure:

1 In the Administration Console, click Access Manager > *SSL VPNs* > *Edit*.

The Server configuration page is displayed.

2 Select *Gateway Configuration* from the *Basic Gateway Configuration* section.

The SSL VPN Gateway Basic Configuration page is displayed.



☐ Behind NAT
External IP Address: N/A
Listening IP Address: 164.99.171.197
Private Address(es): 11.11.11.1
Encryption Port: 7777
Connection Manager Port: 2010
OpenVPN Subnet Address: 10.8.0.0
OpenVPN Subnet Mask: 255.255.255.0
OpenVPN Port: 7777
OpenVPN Protocol: TCP

☐ Provide additional IP Address for OpenVPN
OpenVPN NAT External IP Address: N/A
OpenVPN Listening IP Address: 164.99.171.197

Inactivity Timeout (Minutes): 30
Encryption: AES256
Debug Level: On

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

Fill in the following fields:

- **Behind NAT:** Do not select this check box.
- **External IP Address:** When the *Behind NAT* check box is not selected, this option is disabled.

- **Listening IP Address:** Specify the IP address that SSL VPN listens on. By default, both Enterprise mode as well as Kiosk mode listen on this IP address, but Kiosk Mode listens on the TCP protocol whereas the Enterprise Mode listens on the UDP protocol.
- **Private Address(es):** Specifies the IP address of the private interface of the network card. If you have multiple private networks, specify the private IP addresses of the servers, separated by a comma.
- **Encryption Port:** The port to encrypt traffic. The default encryption port is 7777.
- **Connection Manager Port:** The port on which the connection manager listens to. The default port number is 2010.

  If you change the connection manager port from 2010 to any other port, do the following:

  1. Open `config.txt`, located at `/var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF` and change the port number found in the first line of the file to the new number.
  2. Restart the SSL VPN server manually by entering the following commands:

     ```
     /etc/init.d/novell-sslvpn stop
     ```

     ```
     /etc/init.d/novell-sslvpn start
     ```

  3. Restart Tomcat manually by entering the following commands:

     ```
     /etc/init.d/novell-tomcat4 stop
     ```

     ```
     /etc/init.d/novell-tomcat4 start
     ```

- **OpenVPN Subnet Address:** Specify the IP address for the OpenVPN subnet, which is used with the *OpenVPN Subnet Mask* field to define a pool of addresses that can be dynamically assigned to clients. This information is essential to provide Enterprise mode access.

  NOTE: Make sure that the assigned subnet address and the IP address of the client do not match.

- **OpenVPN Subnet Mask:** Specify the mask for the OpenVPN subnet, which is used to define the address pool. This information is essential to provide Enterprise mode access.

  NOTE: For a given subnet mask:

  The number of client connections = (Available IPs based on subnet mask - 2[0 and broadcast address] - 2[for VPN server and server OS] - 4[only if subnet mask is greater than 29]) / 4

- **OpenVPN Port:** Specify the OpenVPN port number on which the OpenVPN service listens. This information is essential to provide Enterprise mode access.
- **OpenVPN Protocol:** Specify the OpenVPN service protocol. The protocol can either be TCP or UDP. This information is essential to provide Enterprise mode access.
- **Provide Additional IP Address for OpenVPN:** Select this check box if you want to provide an additional listening IP address for SSL VPN in the Enterprise mode. By default, both Kiosk mode as well as Enterprise mode listens on the same IP address and port 777, but use different protocols. If you want both the modes to listen on same the protocol and port, then you can select this check box provide an additional IP address for Enterprise mode to listen in the *OpenVPN Listening IP Address* field. This way, kiosk mode listens on the *Listening IP Address* that you have configured and Enterprise mode listens on the OpenVPN Listening IP Address, while using the same port and protocol.

For example, if you want both Kiosk and Enterprise mode to listen on TCP protocol and port 443, then you can add an additional IP address in the *OpenVPN Listening IP Address* field for Enterprise Mode. In that scenario, Kiosk mode listens on the IP address that you configured in the *Listening IP Address* field, TCP protocol, and port 443; and Enterprise mode listens on the alias IP address that you configured in the *OpenVPN Listening IP Address* field, TCP protocol, and port 443.

NOTE: This is an optional configuration and SSL VPN will continue to operate in both Enterprise mode as well as Kiosk mode, even if this additional IP address is not provided.

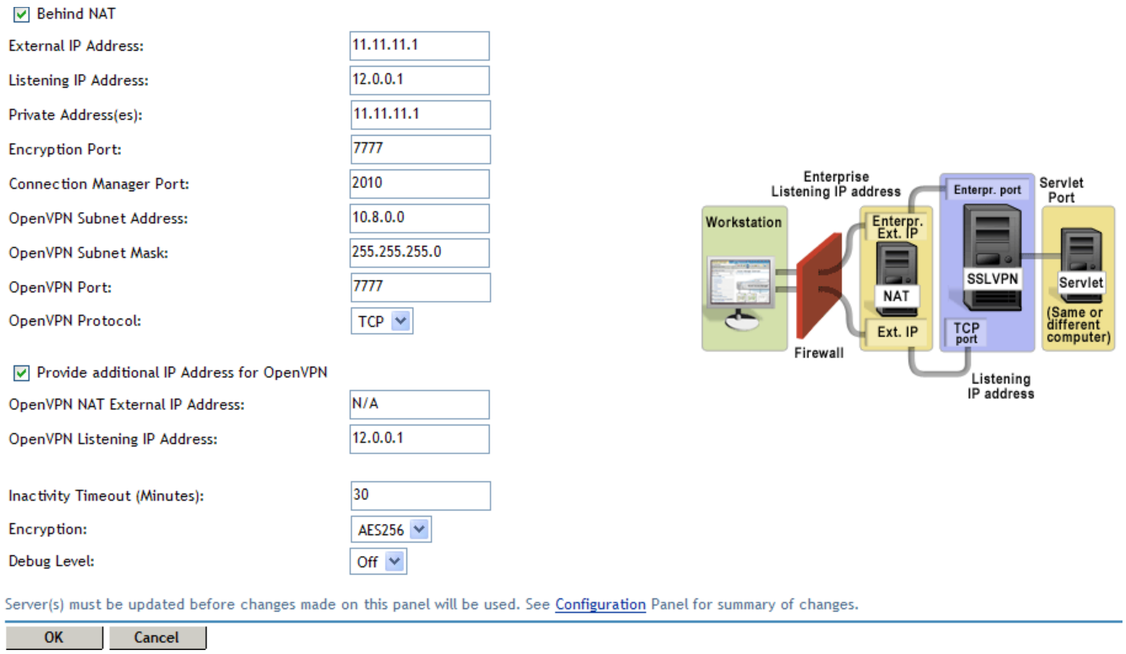
- **OpenVPN Listening IP Address:** This field is enabled if you select the *Provide Additional IP Address for OpenVPN* check box. Specify the additional IP address that the SSL VPN will listen on in the Enterprise mode.
- **OpenVPN NAT External IP Address:** This field is not enabled if the servers are not behind NAT.
- **Inactivity Timeout (Minutes):** Configure the time in minutes after which an idle connection should be closed. If no data exchange takes place during the stipulated time, the connection is closed. An inactive connection is closed after a stipulated time so that the resources are freed to allow additional incoming connections. The inactivity timeout period can be one minute to 1800 minutes. The default inactive timeout period is 30 minutes.
- **Encryption:** Select the type of encryption. It can be either AES 128 or AES 256. AES 256 is the default and recommended encryption mode.
- **Debug Level:** Set this option to *On* if you want more information in the log files. This option is set to *Off* by default. Setting the debug level to *On* helps the administrator in solving any issues with the SSL VPN.

3 To save your modifications, click *OK,* then click *Update* on the Configuration page.

19.2.2 Configuring the SSL VPN Gateway behind NAT

You can configure SSL VPN to be behind NAT. When you do this, you must provide an external IP address, which will be associated with the internal IP address.

Figure 19-1 *Configuring SSL VPN behind NAT*



☑ Behind NAT

External IP Address: 11.11.11.1

Listening IP Address: 12.0.0.1

Private Address(es): 11.11.11.1

Encryption Port: 7777

Connection Manager Port: 2010

OpenVPN Subnet Address: 10.8.0.0

OpenVPN Subnet Mask: 255.255.255.0

OpenVPN Port: 7777

OpenVPN Protocol: TCP

☑ Provide additional IP Address for OpenVPN

OpenVPN NAT External IP Address: N/A

OpenVPN Listening IP Address: 12.0.0.1

Inactivity Timeout (Minutes): 30

Encryption: AES256

Debug Level: Off

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

1 In the Administration Console, click *Access Manager* > *SSL VPNs* > *Edit*.

The Server configuration page is displayed.

2 Select *Gateway Configuration* from the *Basic Gateway Configuration* section.

The SSL VPN Gateway Basic Configuration page is displayed.

You can configure NAT as follows:

- **Behind NAT:** Select the check box to specify that the SSL VPN Gateway is behind NAT.
- **External IP Address:** This field is enabled when the *Behind NAT* check box is selected. Specify the IP address by which the external user on the Internet can access the SSL VPN server.
- **OpenVPN NAT External IP Address:** This field is enabled if you select the *Behind NAT* and *Provide Additional IP Address for OpenVPN* check boxes. Specify the external IP address, when the server is behind NAT in the Enterprise mode. This is an optional configuration and the SSL VPN operates in Enterprise mode as well as Kiosk mode, even if this additional IP address is not provided.

NOTE: For more information on configuring other fields, refer to Section 19.2.1, "Configuring the SSL VPN Gateway without NAT," on page 322.

3 To save your modifications, click *OK,* then click *Update* on the Configuration page.

19.3 Configuring DNS Servers for the Kiosk Mode

The DNS servers configured here are pushed to the client from the SSL VPN server during the connection. You can configure DNS servers for the Enterprise mode of SSL VPN through the Administration Console. The DNS servers can be configured for Kiosk mode either during the installation if you are installing Linux Access Gateway and SSL VPN on the same machine, or by using YaST after the installation. This section has the following information.

The DNS servers can be configured for the Kiosk mode of SSL VPN, either during installation or by using YaST The configuration procedure is dependent on whether you have installed SSL VPN and Linux Access Gateway on the same machine or on separate machines.

NOTE: You must configure the DNS server for both the Kiosk mode and the Enterprise mode. For information on configuring DNS servers for the Enterprise mode, see Section 19.4.2, "Configuring DNS Servers for the Enterprise Mode," on page 327.

This section has the following information:

- Section 19.3.1, "Configuring DNS Servers During Installation," on page 326
- Section 19.3.2, "Configuring DNS Servers After the Installation," on page 326

19.3.1 Configuring DNS Servers During Installation

If you are installing SSL VPN and Linux Access Gateway on the same machine, you can configure DNS Servers during the Linux Access Gateway installation. For more information, see "Installing the Linux Access Gateway" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

19.3.2 Configuring DNS Servers After the Installation

If you are installing SSL VPN and the Linux Access Gateway on separate machines, you can configure DNS servers in the `/etc/resolv.conf` file by using YaST as follows:

1. In YaST, select *Network Devices* > *Network Cards*, then press Enter.
2. Select *Change*, then press Enter.
3. Select *Edit*, then press Enter.
4. Select *Hostname and Name Servers*, then press Enter.
5. Specify the IP addresses of the DNS servers that you want to add.
6. Specify the domain names.
7. Click *OK*.

Verify that the DNS servers and Domain names are added to the `/etc/resolv.conf` file.

19.4 Additional Configuration for Enterprise Mode

The Enterprise mode of SSL VPN requires some additional configurations to be done through the command line.

- Section 19.4.1, "Configuring Routes," on page 327
- Section 19.4.2, "Configuring DNS Servers for the Enterprise Mode," on page 327

19.4.1 Configuring Routes

In Enterprise mode, SSL VPN assigns IP addresses to each client from subnet specified in the configuration. For more information on configuring IP address, see Section 19.2, “Configuring the IP Address, Port, and NAT,” on page 322. The values specified in the *OpenVPN Subnet Address* and *OpenVPN Subnet Mask* fields determine the IP addresses that are assigned to the clients. Make sure that the assigned IP address and the IP address of the client do not match.

The packets from these clients reach the application server with the IP address of the client as the source address. The response packets need to be routed back to the SSL VPN server, which sends them on to the clients. You can solve this routing problem in one of the following ways:

- “Configuring the OpenVPN Subnet in Routing Tables” on page 327
- “Configuring Source NAT” on page 327

Configuring the OpenVPN Subnet in Routing Tables

If you have a gateway for your network between the application server and the SSL VPN server, you can configure the gateway to send the dynamically assigned IP addresses from the OpenVPN address pool to the SSL VPN server. This is the best routing approach because most applications, including ActiveFTP and TFTP, can work in this type of environment. To establish this type of routing, you need to add a static route to your network’s routing infrastructure so that traffic to the OpenVPN subnet pool of addresses is sent via the SSL VPN gateway.

Configuring Source NAT

You can configure Source NAT to change the dynamically assigned client addresses to the address of the SSL VPN server before sending them to the application server. The application server can then use the source address in the packets to send them back to the SSL VPN server, which can then reassign the client address and send the packets on to the client. This is the best approach if you are using SSL VPN for TCP and UDP applications. Other applications, such as ActiveFTP and TFTP, cannot work in this type of environment. To establish this type of routing, you need to create an entry in the `iptables` file on the SSL VPN server. If the *OpenVPN Subnet Address* option is set to 10.8.0.0/16 and the IP address of the SSL VPN server is 10.16.12.247, the entry should be similar to the following:

```
iptables -t nat -A POSTROUTING -s 10.8.0.0/16 -j SNAT --to 10.16.12.247
```

Restart the SSL VPN services after the `iptables` file has been modified.

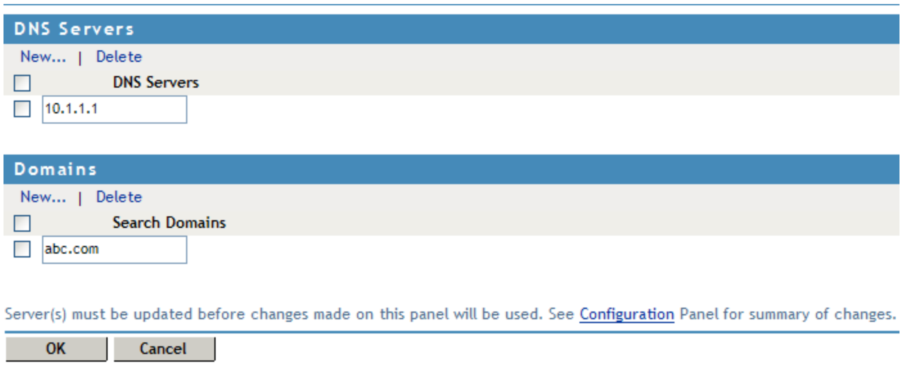
IMPORTANT: This simple solution only works if you are not using iptables to translate ports of other applications or Access Manager components. For a solution that works with multiple components, see Configuring SUSE Firewall for the SSL VPN Component in Access Manager (http://www.novell.com/coolsolutions/appnote/19939.html).

19.4.2 Configuring DNS Servers for the Enterprise Mode

1. In the Administration Console, click *Access Manager* > *SSL VPNs* > *Edit*.

   The Server configuration page is displayed.
2. Select *DNS Server List* from the *Basic Gateway Configuration* section.

   The DNS server list page is displayed.



☐ 10.1.1.1

☐ abc.com

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

3 To configure a DNS server, click *New* in the *DNS server* section, specify the IP address of the server, then click *OK*.

4 To configure a Domain, click *New* in the *Domains* section, specify the domain name, then click *OK*.

5 To delete a DNS server or a domain, select the check box next to the field and click *Delete* in the section.

6 To save your modifications, click *OK,* then click *Update* on the Configuration page.

Controlling Access

20

The Novell® Access Manager SSL VPN allows you to configure traffic policies to control access to resources based on the role of the client. The traffic policies are a set of rules and regulations, administered to regulate user access to the protected network resources. Novell SSL VPN traffic policies are role-based policies. The access to the protected network is restricted based on the role to which the user belongs.

This section contains the following information:

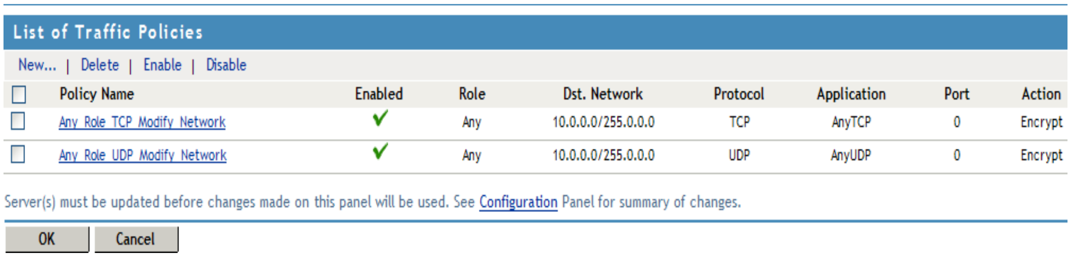
- Section 20.1, "Configuring Traffic Policies," on page 329
- Section 20.2, "Configuring Client Integrity Check Policy to Protect the Internal Network," on page 331

20.1 Configuring Traffic Policies

You can configure a maximum of 250 traffic rules per role, depending on the length of the policy name. If you have configured multiple traffic policies, the policies are prioritized based on the order of their creation.

You can configure a different set of traffic policies for different roles as follows:

1. In the Administration Console, click *Access Manager* > *SSL VPNs* > *Edit*.
2. Select *Traffic Policies* from the *Policies* section. The SSL VPN Traffic Policies page is displayed.

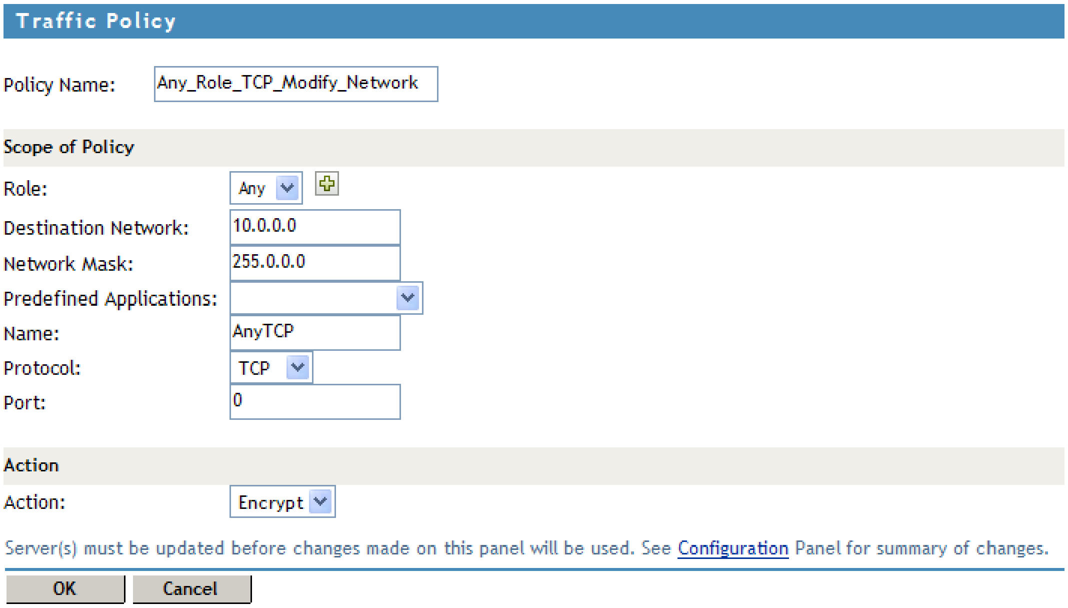


List of Traffic Policies

☐	Policy Name	Enabled	Role	Dst. Network	Protocol	Application	Port	Action
☐	Any_Role_TCP_Modify_Network	✓	Any	10.0.0.0/255.0.0.0	TCP	AnyTCP	0	Encrypt
☐	Any_Role_UDP_Modify_Network	✓	Any	10.0.0.0/255.0.0.0	UDP	AnyUDP	0	Encrypt

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

3. Click *New*. The New dialog box is displayed.
4. Specify the traffic policy name in the *Traffic Policy Name* field, then click *OK*.
5. Click the newly added traffic policy. The Edit Traffic Policy page is displayed.



Policy Name: Any_Role_TCP_Modify_Network

Scope of Policy

Role: Any

Destination Network: 10.0.0.0

Network Mask: 255.0.0.0

Predefined Applications:

Name: AnyTCP

Protocol: TCP

Port: 0

Action

Action: Encrypt

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

Fill in the following fields:

- **Policy Name:** Specify the name for the traffic policy.
- **Role:** The role to which the traffic rule applies. Select the role from the drop-down list. If the role is not listed, click the role icon to add new roles.

  The Role is case-sensitive. If the role configured is `Employee` and the Identity Server sends a request for `employee`, the rule is not pushed to the client.
- **Destination Network:** Specify the host or network IP address of the destination.
- **Network Mask:** The network mask is displayed by default when you specify the destination address. However, you can edit the mask.
- **Predefined Application:** Select a predefined application from the drop-down list.
- **Name:** Specify a name for the application. This information is optional.
- **Protocol:** Select a protocol from the drop-down list. You can select the protocol to be TCP, UDP, ICMP or Any.
- **Port:** Specify the port number on which the service you select must listen.

  NOTE: Specify 0 to allow all ports depending on the protocol.
- **Action:** Specify if a service can be allowed or denied. Select *Encrypt* to allow the service in encrypted form. Select *Deny* if you do not want to allow the service.

6 To delete a traffic rule, select the rule, then click *Delete*.

7 To enable a traffic rule, select the rule, then click *Enable*.

8 To disable a traffic rule, select the rule, then click *Disable*.

9 To save your modifications, click *OK*, then click *Update* on the Configuration page.

20.2 Configuring Client Integrity Check Policy to Protect the Internal Network

Novell SSL VPN has a set of client integrity check policies to protect your network and applications from clients that are using insufficient security restraints. You can configure a client integrity check policy to run on the client workstations before establishing a tunnel to the SSL VPN gateway. The check ensures that the users have specified software installed and running in their systems.

- Section 20.2.1, "Overview of Client Integrity Check Policies," on page 331
- Section 20.2.2, "Configuring the Client Integrity Check Policy," on page 332

20.2.1 Overview of Client Integrity Check Policies

You can configure the client integrity check policy to check for application categories such as Firewall, Antivirus, and Mail clients depending on your requirements. The client integrity check policy is configured in the following stages:

1. **Configure Category:** A category is a group of similar software. For example, a firewall category can contain a list of firewall such as Windows Firewall and Zone alarm firewall. You can configure multiple software categories in the client integrity check policy. The client workstation is checked to see if the software specified under these categories is installed in the workstation, before the SSL VPN connection is established.
2. **Configure Applications Names for a Category:** After you have created a category, you must add application names to that category. An application name is the name of the software configured under a particular category. You can add more than one software item under a category. A client workstation is checked for the presence of any one of the software items in the category. If none of the software specified in the category is present, then the client integrity check fails and the tunnel to the SSL VPN gateway is not established.
3. **Configure Application Details:** After you have added an application to a category, you must configure the attributes of that particular application. The following table lists the attributes for applications on different operating systems:

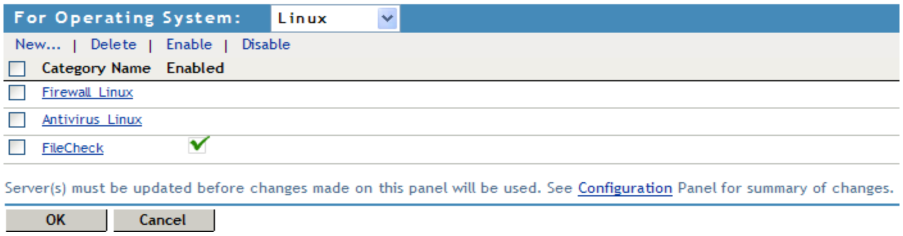
Operating System	Attribute Type	Attribute Name
Linux	RPM	**Name:** Specifies the name of the RPM.
		Version: Specifies the RPM version.
	Process	**Name:** Specifies the name of the process.
		Owner: Specifies the owner of the process.
	Absolute File	**Name:** Specifies the name and absolute path of the file.

Operating System	Attribute Type	Attribute Name
Windows	Process	**Name:** Specifies the name of the executable file if the application is a process, **Version:** Specifies the software version. **RegistryKey:** Specifies the registry key path. **RegistryKeyValue:** Specifies the registry key value. The value data found in this key value should be the absolute path of the folder where the process file is present.
	RegistryKey	**Name:** Specifies the name of the RegistryKey.
	Absolute File	**Name:** Specifies the name of the absolute path of the file name. **Version:** Specifies the owner of the process.
Macintosh	Package	**Name:** Specifies the name of the software package. **Version** Specifies the version of the software package
	Process	**Name:** Specifies the name of the process **Owner:** Specifies the owner of the process.
	Absolute File	**Name:** Specifies the name of the executable file if the application is a process,

20.2.2 Configuring the Client Integrity Check Policy

To configure a client integrity check policy for SSL VPN:

1. In Administration Console, click *Access Manager > SSL VPNs > Edit*.
2. Select *Client Integrity Check Policies* from the *Policies* section. The Client Integrity Check Policies page is displayed.

   
   For Operating System: Linux
New... | Delete | Enable | Disable
OK
Cancel

☐ Category Name	Enabled
☐ Firewall_Linux	
☐ Antivirus_Linux	
☐ FileCheck	✓

   Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

   You can perform the following actions in this page:

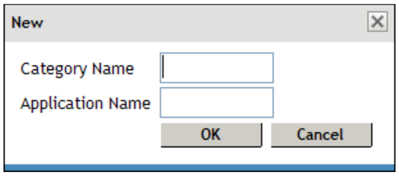
   New: To add a new category, click *New*.

   Delete: To delete a category, select the category, then click *Delete*.

   Enable: To enable a category, select the check box next to category, then click *Enable*. The newly added category is enabled by default.

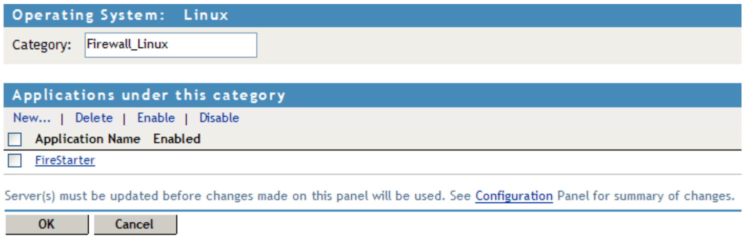
   Disable: To disable a category, select the check box next to category, then click *Disable*.
3. Select the operating system from the *Operating System* drop-down list.

4 Click *New* to enter a new software category. The New dialog box is displayed.



5 Specify a *Category Name* and an *Application Name*, then click *OK*.

6 Click the newly added category to add applications to it. The Client Integrity Check - Category page is displayed.


Operating System: Linux

Category: Firewall_Linux

Applications under this category
New... | Delete | Enable | Disable
☐ Application Name Enabled
☐ FireStarter

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

You can perform the following actions on this page:

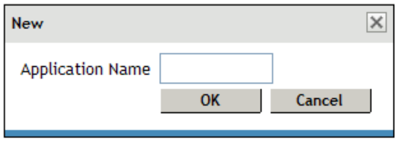
New: To add a new application to the category, click *New*.

Delete: To delete an application, select the application, then click *Delete*.

Enable: To enable an application, select the check box next to application, then click *Enable*.

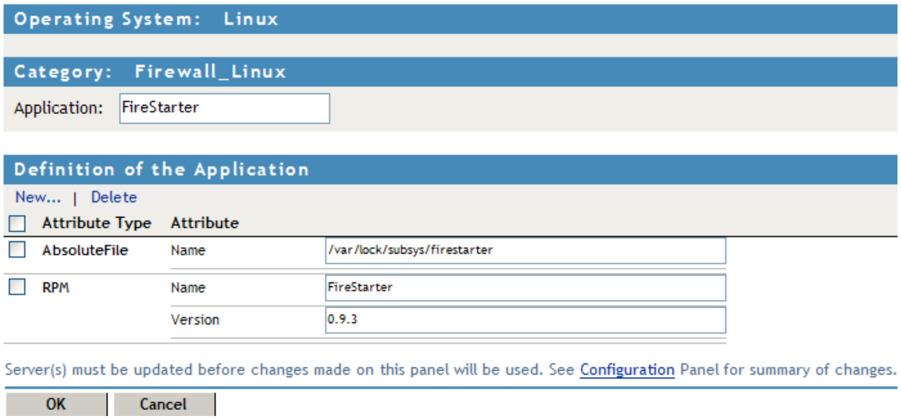
Disable: To disable an application, select the check box next to application, then click *Disable*.

7 Click *New* to add a new application to the category. The new dialog box is displayed.



8 Specify an application name, then click *OK*.

9 Click the newly added application to add application details and attributes to it. The Application Details and Attributes page is displayed.


Operating System: Linux

Category: Firewall_Linux

Application: FireStarter

Attribute Type	Attribute	
☐ AbsoluteFile	Name	/var/lock/subsys/firestarter
☐ RPM	Name	FireStarter
	Version	0.9.3

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

This page specifies the operating system, application category and name and details of the application. The following application details are listed in this page:

- **Attribute Type:** Specifies whether the attribute is a Process, Package, AbsoluteFile, Registry Key, or an RPM, based on the type of operating system you select.
- **Attribute Name:** Specifies attribute names for different attribute types. For more information, see "Overview of Client Integrity Check Policies" on page 331.
- **Attribute Value:** Specifies the value of each attribute name.

10 Click *New* to add an attribute to the application, add an attribute name, then click *OK*.

11 Click *OK,* to save your modifications, then click *Update* on the Configuration page.

Managing Server Settings

21

This section describes how to modify the SSL VPN server details, moving the SSL VPN server to a different Administration Console, and configuring SSL VPN to download the Java applet on Internet Explorer.

- Section 21.1, "Advanced Configuration Settings," on page 335
- Section 21.2, "Configuring SSL VPN to Connect through Forward Proxy," on page 339
- Section 21.3, "Configuring Load Balancing and Fault Tolerance," on page 340
- Section 21.4, "Configuring Certificate Settings," on page 343
- Section 21.5, "Modifying SSL VPN Server Details," on page 345
- Section 21.6, "Moving the SSL VPN Server to a Different Administration Console," on page 346

21.1 Advanced Configuration Settings

This section discusses some of the advanced setting that can be configured for the SSL VPN servers.

- Section 21.1.1, "Configuring SSL VPN to Download the Applet on Internet Explorer," on page 335
- Section 21.1.2, "Configuring SSL VPN to Connect Only in Kiosk Mode," on page 336
- Section 21.1.3, "Customizing the SSL VPN Home Page," on page 336
- Section 21.1.4, "Configuring Full Tunneling for the Kiosk Mode," on page 338
- Section 21.1.5, "Creating DH Certificates with Different Key Sizes," on page 338

21.1.1 Configuring SSL VPN to Download the Applet on Internet Explorer

The SSL VPN client components are carried forward to client desktop through Java applet or ActiveX, along with the policies and the required client components.

Some Windows clients do not allow ActiveX controls to run in the Internet Explorer. In such scenarios, the user can force the Windows client to load a Java-based applet instead of the ActiveX controls. In order to force load the applet, enter the following URL to launch the SSL VPN user interface:

https:<*DNS-Name*>/sslvpn/login?forcejre

If your company's policy does not allow ActiveX controls to be downloaded through Internet Explorer, you can change the SSL VPN configuration to always download the applet-based client. You can change the value within the `<param-value>` tags in the `web.xml` file to `true` from `false` as follows:

1. Log in as `root`.
2. Open the `web.xml` file found in the following location:

   `/var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF/`

3 In the `<context-param>` section, change the `<param-value>` to `true` as follows:

```
<context-param>
<param-name>forcejre</param-name>
<param-value>true</param-value>
<description>My organization does not allow activex ? enter true
if so</description>
</context-param>
```

Save the `web.xml` file.

4 Restart the Tomcat server by entering the following command:

```
/etc/init.d/novell-tomcat restart
```

21.1.2 Configuring SSL VPN to Connect Only in Kiosk Mode

You can configure SSL VPN to connect in Kiosk mode only, even if the user is the admin or `root` user of the machine. To configure SSL VPN to connect in Kiosk mode only, update the `config.txt` file as follows:

1 Login as `root`.

2 Open `config.txt` which is located in the following path:

```
/var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF/
```

3 Append the following line to the file:

```
ForceKiosk=true
```

4 Save and close the file.

5 Enter the following command to restart tomcat:

```
/etc/init.d/novell-tomcat4 restart
```

NOTE: By default, you cannot enable the Kiosk mode only option to Windows non-admin users connecting to SSL VPN through the Internet Explorer. This is because admin user privilege is required to download the ActiveX component of SSL VPN. But if the user's Windows machine has JRE installed, then you can force Internet Explorer to connect to SSL VPN through applet. For more information about this, see Section 21.1.1, "Configuring SSL VPN to Download the Applet on Internet Explorer," on page 335.

21.1.3 Customizing the SSL VPN Home Page

You can customize the contents of the SSL VPN home page and the company logo depending on the requirements of the organization. This section has the following information:

- "Changing the Logo" on page 336
- "Customizing the Content of Home Page" on page 337
- "Customizing the SSL VPN UI in Applet" on page 337

Changing the Logo

1 Browse to the following location and replace `SSLVPN_Nlogo.gif`

```
/var/opt/novell/tomcat4/webapps/sslvpn/pages/other
```

You must retain the filename and file size of the original graphic.

NOTE: In the localized versions, the `.gif` file is located in the `/pages_<language>/other` folder. For example, if you want to customize the German version, you must browse to the following location:

```
/var/opt/novell/tomcat4/webapps/sslvpn/pages_de/other
```

---

2 In ActiveX, the logo is hyperlinked to www.novell.com. To change the hyperlink:

  2a Open `/var/opt/novell/tomcat4/webapps/sslvpn/pages/banner.html`.

  2b Browse to the `<div id=”logo”>` section, then change the `<HREF>` link to the URL of your choice.

  2c Save and close the file.

Customizing the Content of Home Page

1 Browse to `/var/opt/novell/tomcat4/webapps/sslvpn/pages`.

2 Do the following:

  2a Modify the contents of `home.html` file. This file is displayed to the user when ActiveX is downloaded to the client machine.

  2b Modify the contents of `pre_applet_home.html` and `applet_home.html`. These files are displayed to the user when a Java applet is downloaded to the client machine. The contents of `pre_applet_home.html` is displayed to the user when the SSL VPN connection is being made. This page changes to `applet_home.html` page, after the connection status changes to *Connected*.

3 Save and close the file.

NOTE:
- In the localized versions, `banner.html` is located in the `/pages_<language>` folder. For example, if you want to customize the German version, you must browse to the following location:

  ```
  /var/opt/novell/tomcat4/webapps/sslvpn/pages_de
  ```
- If you have referenced other Web pages from your home page, you must wait till the connection status changes to *Connected*, before clicking the link.

Customizing the SSL VPN UI in Applet

You can customize the height and width of the UI and the width of the tabs used in the SSL VPN user interface.

1 Log in as `root`.

2 Open `config.txt` which is located in the following path:

```
/var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF/
```

3 Specify the following lines:

```
UIConfig=768, 1000, 150
1st param = Height of the applet UI
2nd param = Width of the applet UI
```

```
3rd param = Width of the tabs list (which contains Home,
policies, logs etc.)
```

UIConfig=768, 1000, 150 is the default look configuration. Some of the other parameters that can be used are:

```
smart look = 900,1200,100
full look = 1000,1400,90
Overloaded look = 1200,1600,150
```

4. Save and close the file.

21.1.4 Configuring Full Tunneling for the Kiosk Mode

Novell SSL VPN is configured for split tunneling by default. When SSL VPN is configured for split tunneling, only that traffic that is destined for the protected network goes through the VPN tunnel. However, if you have connected to the SSL VPN in the Kiosk mode and you want all traffic in the client machine to go through the tunnel (full tunneling), do the following:

1. In the Administration Console, click *Access Manager* > *SSL VPNs* > *Edit*.
2. Create a new traffic policy. For more information on adding new traffic policy, see Section 20.1, "Configuring Traffic Policies," on page 329.
3. Click the newly added traffic policy. The Edit Traffic Policy page is displayed.

   Configure the following fields:

   - **Destination Network:** Specify 0.0.0.0 as the destination network IP address.
   - **Protocol:** Select *Any* as the protocol.
   - **Port:** Specify the port number as 0.
   - **Action:** Select *Encrypt* to allow the service in encrypted form.

   Leave the default values in the other fields unchanged.
4. Click *OK* to save changes.
5. In the Edit page, select *Gateway Configuration* from the *Basic Gateway Configuration* section.

   The SSL VPN Gateway Basic Configuration page is displayed.
6. In the *Private IP Address(es)* field, specify all the IP addresses that the SSL VPN server can use to access the public resources.
7. To save your modifications, click *OK*, then click *Update* on the Configuration page.

NOTE: Full tunneling is not supported in the Enterprise mode.

21.1.5 Creating DH Certificates with Different Key Sizes

The Enterprise mode of SSL VPN uses DH certificates for encryption. These certificates are created automatically during the installation or upgrade, with a default key size of 1024. You can create DH certificates with key sizes of your choice. You can create a DH certificate with a maximum key size of 4096. To create a DH certificate with a key size of your choice, enter the following command:

```
sslvpnc -k <keysize>
```

Replace <*keysize*> with the key size of your choice.

21.2 Configuring SSL VPN to Connect through Forward Proxy

The Novell SSL VPN can be configured to detect and connect to forward proxy in both Kiosk as well as Enterprise modes after authenticating to the Identity server. To establish the SSL VPN connection through forward proxy, you can either configure the browser or create a `proxy.conf` file in the user's home directory. This file must contain the IP address and the port number of the forward proxy in the following format:

`proxyHost=<IPaddress>:<port number>`

For example: `proxyHost=172.10.0.0:8080`

NOTE: If you want to establish the SSL VPN connection in the Enterprise mode through forward proxy, the server must be listening on the TCP port and not on the UDP port.

When a user initiates a connection to SSL VPN server through a browser:

1. SSL VPN checks if the browser uses a proxy.
2. If yes, SSL VPN checks for the proxy configuration file `proxy.conf` in the user's home directory. The `proxy.conf` file must have the IP address and the port number of the forward proxy entered in the following format:
3. If a proxy configuration file is present, the following occurs:
    a. It checks for the format of the file. If the information provided in the file is not in the correct format, then SSL VPN proceeds with Step 4.
    b. If the configuration information provided is in the correct format, SSL VPN reads the proxy information from the `proxy.conf` file, then proceeds with Step 6.
4. If the proxy configuration file is not present, SSL VPN checks for proxy configuration information from the browser registry or profile.
5. If SSL VPN is unable to get the proxy configuration information either through the `proxy.conf` file or through the registry, it throws an error asking the user to edit the `proxy.conf` and terminates the connection process.
6. SSL VPN reads the proxy configuration information and attempts to connect to the resource without the proxy. If this attempt fails, then the SSL VPN connection is made through the forward proxy.

NOTE: If authentication is enabled in forward proxy, SSL VPN in Kiosk mode will not be able to connect through forward proxy. But you can establish SSL VPN connection in Enterprise mode through forward proxy if the authentication method used is basic or NTLM. For more information on how to connect to forward proxy with authentication, see Section 21.2.1, "Connecting to Forward Proxy with Authentication in Enterprise Mode," on page 340. We do not recommend this method as you need to specify the credentials of forward proxy in the configuration file and this might be a security vulnerability.

21.2.1 Connecting to Forward Proxy with Authentication in Enterprise Mode

To connect to forward proxy with authentication enabled, you must create an authentication file and save the username and password of the forward proxy. The `proxy.conf` file is then updated with the name of this authentication file.

NOTE: This procedure works in Linux and Macintosh environments only.

---

1. Create an authentication file with the username and password in the following format:

   ```
   <username>
   <password>
   ```

2. Save the file in the `<name>.auth` format.

   Replace <*name*> with a file name of your choice.

3. Modify the `proxy.conf` file as follows:

   ```
   proxyHost=<IPaddress>:<port number> <authfile> <auth-method>
   ```

   Replace <*authfile*> with the name and the path to the authentication file and replace <*auth-method*> with the authentication method.

   For example, `proxyHost=172.10.0.0:8080 c:\abc.auth basic`

4. Save the `proxy.conf` file.

21.3 Configuring Load Balancing and Fault Tolerance

SSL VPN enables configuration of server failover groups, which enable load balancing and fault tolerance. These groups ensure that when a server goes down, the other servers can service the clients. The following sections describe procedure to configure load balancing and fault tolerance for SSL VPN.

- Section 21.3.1, "Configuring Load Balancing Through the Access Gateway," on page 340
- Section 21.3.2, "Configuring Load Balancing Through Servlets," on page 342

21.3.1 Configuring Load Balancing Through the Access Gateway

You can install and run the SSL VPN self-monitoring and failover scripts on each SSL VPN server in order to provide automatic monitoring and failover support for the SSL VPN servers that are behind either a NetWare® Access Gateway or a Linux Access Gateway.

When the health status of an SSL VPN server is bad, these scripts modify the IPTables entries on that server to stop the Access Gateway from sending connection requests to that particular SSL VPN server. When the SSL VPN server health status returns to normal, the scripts remove the IPTables entries and allow the Access Gateway to communicate with the SSL VPN server. You must do the following to configure load balancing and fault tolerance through access gateway:

1. Configuring the Access Gateway.
2. Installing the Scripts

3. Testing the Scripts

Configuring the Access Gateway

1. In the Administration Console, click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Web Servers*.
2. Add all the SSL VPN servers that are part of the failover group as origin Web servers to the proxy service that you have defined.
3. Click *TCP Connect Options*.
4. Select *Round Robin* in the *Policy for Multiple Destination IP Addresses* field.
5. Select *Enable Persistent Connections*.
6. Save your changes and update the Access Gateway.

Installing the Scripts

1. Download the tar file containing Scripts for SSL VPN Automatic Monitoring and Failover from the Additional Resources section in the Novell Access Manager documentation page (http://www.novell.com/documentation/novellaccessmanager/index.html). The tar file contains `sslvpn-heartbeat.sh` and sslvpn-heartbeat.
2. Copy the `sslvpn-heartbeat.sh` script to the `/opt/novell/sslvpn/bin` directory in each of the SSL VPN servers.
3. Copy the `sslvpn-heartbeat` file to the `/etc/init.d/`directory.
4. Enter the following commands to change `sslvpn-heartbeat.sh` and `sslvpn-heartbeat` into executable files:

   ```
   chmod +x sslvpn-heartbeat.sh
   ```

   ```
   chmod +x sslvpn-heartbeat
   ```
5. Enter the following command to run the script every time the Access Gateway is started:

   ```
   insserv /etc/init.d/sslvpn-heartbeat
   ```

Testing the Scripts

1. Enter the following command to stop the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn stop
   ```
2. Enter the following command to verify if the scripts have blocked port 8080:

   ```
   iptables -L
   ```

   The following lines are displayed if port 8080 is blocked:

   ```
   Chain       sslvpn-heartbeat-chain (1 reference)
   target      prot opt source   destination
   REJECT      tcp  --  anywhere anywhere    tcp
   dpt:http-alt reject-with icmp-port-unreachable
   ```
3. In the Administration Console, click *Access Gateways* > *[Name of Server]* > *Health*. The following message is displayed if the SSL VPN server is down:

   ```
   The HTTP Reverse Proxy service <reverse proxy name> might not be
   functioning properly. Few of the Web servers being accelerated are
   unreachable <sslvpn server IP Address>:8080
   ```

Click *Update from Server* to get the latest health status of the Access Gateway.

4. Connect to SSL VPN. Verify that your connection was sent to the SSL VPN that is running and not to the one that is marked as down by the Access Gateway.
5. Enter the following command to start the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn start
   ```
6. Enter the following command to verify if the script has removed the block on port 8080:

   ```
   iptables -L
   ```

   The following lines are displayed if the block on port 8080 is removed:

   ```
   Chain sslvpn-heartbeat-chain (1 references)
   target     prot opt source    destination
   ```
7. In the Administration Console, click *Access Gateways* > *[Name of Server]* > *Health*, then check that the SSL VPN server is up.

   Click *Update from Server* to get the latest health status of the Access Gateway.
8. Connect to SSL VPN. Verify if your connection was sent to the SSL VPN server that was restarted. It might require several attempts before you can connect to the desired Access Gateway.
9. Repeat Step 1 to Step 8 to verify if the SSL VPN health scripts are working on all the SSL VPN servers.

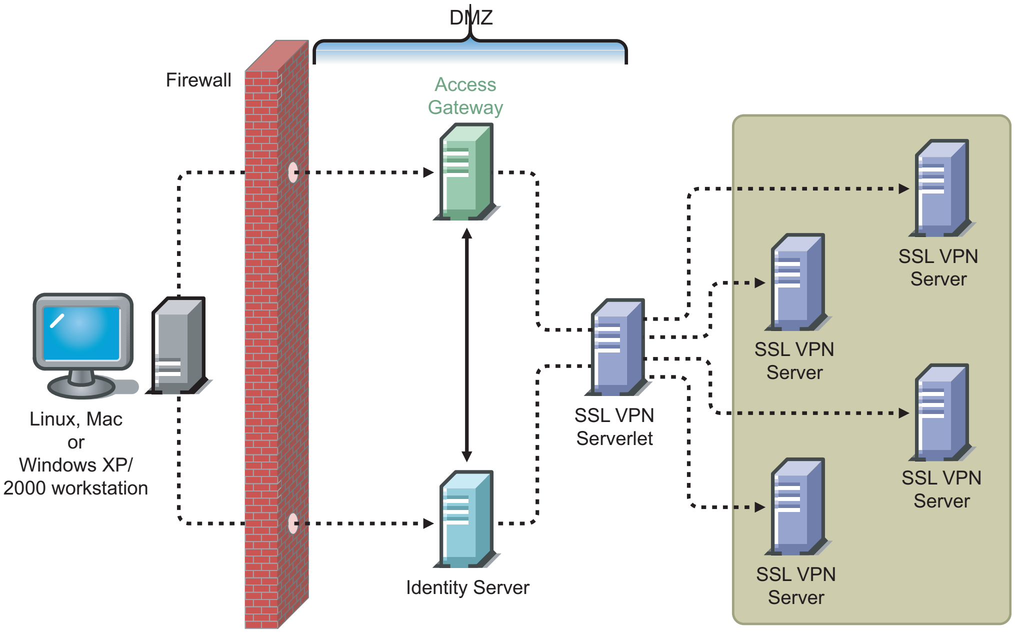
21.3.2 Configuring Load Balancing Through Servlets

The SSL VPN server has load balancing capabilities so that more than one SSL VPN server can handle client connections. You can configure load balancing and fault tolerance on these servers by using the `config.txt` file.

However, it is a passive fault tolerance because if a server goes down, all the client connections to that server are disconnected. When these clients try to reconnect, they are redirected to other servers in the failover group.

You can configure servers in the failover group receive connection. This way, client connections are distributed among the servers of the failover group, thereby balancing the load. You can also configure the servers in such a way that all the client connections are received by one server and when that server goes down, all the connections are redirected to the next server.

Figure 21-1 Load Balancing SSL VPN servers



Download and install the SSL VPN servlet RPM on a separate machine. Modify the `config.txt` file as follows:

1. Open `config.txt` which is located in the following path:

   ```
   /var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF/
   ```

2. The first line of `config.txt` contains the IP address and port number of the default server in the following format:

   ```
   ServerIP=IPaddress:Port=Port number
   ```

   NOTE: Add the IP address and port number of the servers in the same format in the next line. You can add a maximum of four servers to the failover group.

3. To enable load balancing among servers, set `RoundRobinCluster=true`

   If you set the option to `false`, only fault tolerance is enabled and load balancing is not enabled, because all the client connections are received by one server and when that server goes down, all the connections are redirected to the next server

4. Save and close the file.
5. Restart the server by entering the following command:

   ```
   /etc/init.d/novell-tomcat4 restart
   ```

6. Restart all the SSL VPN servers by entering the following command:

   ```
   /etc/init.d/novell-sslvpn restart
   ```

21.4 Configuring Certificate Settings

Access Manager components and agents can access the keystore to retrieve certificates, keys, and trusted roots as needed.

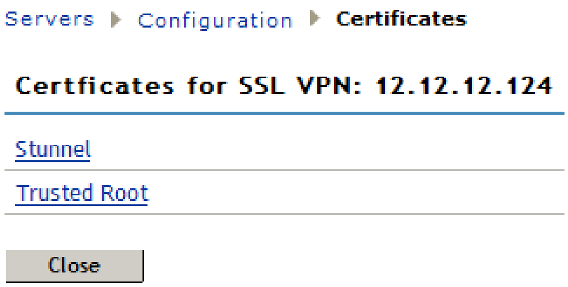
Before you proceed with this section, make sure you have already created a certificate. For more information on creating certificates, see Part V, “Security and Certificate Management,” on page 353.

- Section 21.4.1, “Adding Certificates to the SSL VPN Keystore,” on page 344
- Section 21.4.2, “Adding Trusted Roots for SSL VPN,” on page 345

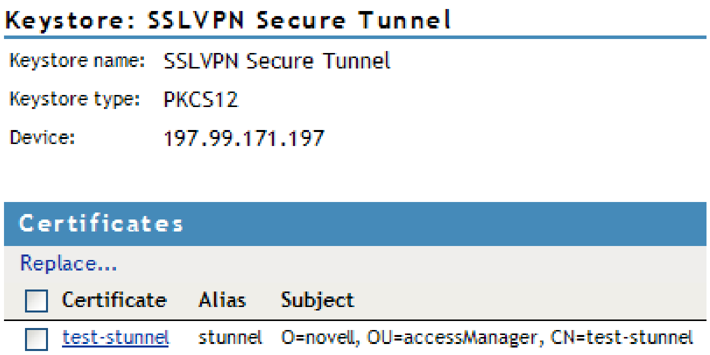
21.4.1 Adding Certificates to the SSL VPN Keystore

1 In the Administration Console, select *Access Manager > SSL VPN > Edit*.

2 Select *SSL VPN Certificates* from the *Security settings* section. The Certificates for SSL VPN page is displayed.


Servers ▸ Configuration ▸ Certificates
Certficates for SSL VPN: 12.12.12.124
Stunnel
Trusted Root
Close

3 Click *STunnel*. The Keystore: SSL VPN Secure Tunnel page is displayed.



Keystore: SSLVPN Secure Tunnel

Keystore name: SSLVPN Secure Tunnel
Keystore type: PKCS12
Device: 197.99.171.197

Certificates

Replace...

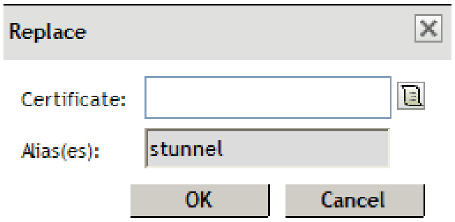
☐ Certificate	Alias	Subject
☐ test-stunnel	stunnel	O=novell, OU=accessManager, CN=test-stunnel

Certificates in the SSL VPN STunnel are used by SSL VPN services for encryption. This page contains the following information:

- **Keystore name:** Specifies the name of the keystore to which the certificate belongs.
- **Keystore type:** Specifies the type of keystore. It can be Java, PEM, or PKCS12
- **Device:** Specifies the IP address of the SSL VPN device.

NOTE: Every imported SSL VPN device has a default certificate.

4 To replace the default certificate, click *Replace*. The Replace dialog box is displayed.



Certificate:
Alias(es): stunnel

Fill in the following fields:

- **Certificates:** Click the *Select Certificate* icon to browse and select the certificate that you want to associate with SSL VPN.
- **Alias(es):** You can provide an alternate name for the certificate you are importing.

5 Click *OK* to save changes.

6 To save your modifications, click *OK,* then click *Update* on the Configuration page.

21.4.2 Adding Trusted Roots for SSL VPN

A trust store contains certificates from a certificate authority (CA). These certificates are self-signed and are recognized as representing a CA that is trusted. When creating a trust store, you can assign trust stores to devices and add trusted root certificates to the new trust stores.

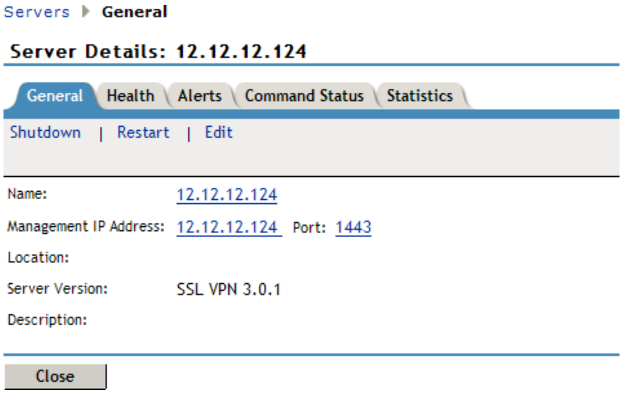
NOTE: Trusted roots need not be configured for SSL VPN.

21.5 Modifying SSL VPN Server Details

To edit the Gateway information:

1 In the Administration Console, click *Access Manager > SSL VPNs > [Server Name]*.

The Server Details page is displayed.

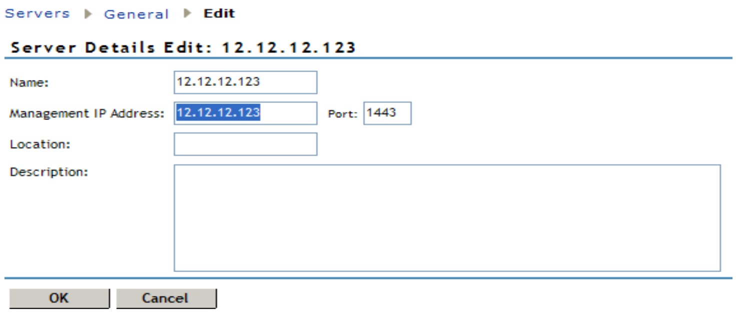


Server Details: 12.12.12.124

Name: 12.12.12.124
Management IP Address: 12.12.12.124 Port: 1443
Location:
Server Version: SSL VPN 3.0.1
Description:

The *General* tab of the Server Details page displays information such as name, Management IP address, Port, Location, and the server version of the selected server.

2 Click *Edit*. The Server Details Edit page is displayed.



Server Details Edit: 12.12.12.123

Name: 12.12.12.123

Management IP Address: 12.12.12.123 Port: 1443

Location:

Description:

You can edit the information in the following fields:

- **Name:** Specify the IP address of the server. This field is mandatory.
- **Management IP Address:** Specify the IP address used to manage the server. If the system on which the agent is installed has multiple IP addresses, you can select one from the drop-down list.
- **Port:** Specify the port used for management. This field is mandatory.
- **Location:** Specify the location of the SSL VPN server.
- **Description:** (Optional) You can provide a brief description of the purpose of this SSL VPN Gateway or any other relevant information.

3 Click *OK* to save changes or *Cancel* to discard the changes.

21.6 Moving the SSL VPN Server to a Different Administration Console

If the SSL VPN Gateway must be moved to a different Administration Console, use the following procedure:

1. In the Administration Console, delete the existing server
2. Enter the following command to stop the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn stop
   ```
3. Enter the following command:

   ```
   sslvpnc --configure
   ```
4. Specify the IP Address of the new administration console.
5. Specify the public IP address of SSL VPN server.
6. Specify the private IP address of SSL VPN server.
7. At the console shell, access the `/opt/novell/devman/jcc` directory, then enter the following command:

   ```
   conf/Configure.sh
   ```
8. Enter the following command to start the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn start
   ```
9. Restart the Access Manager Server Communications by using the following command:

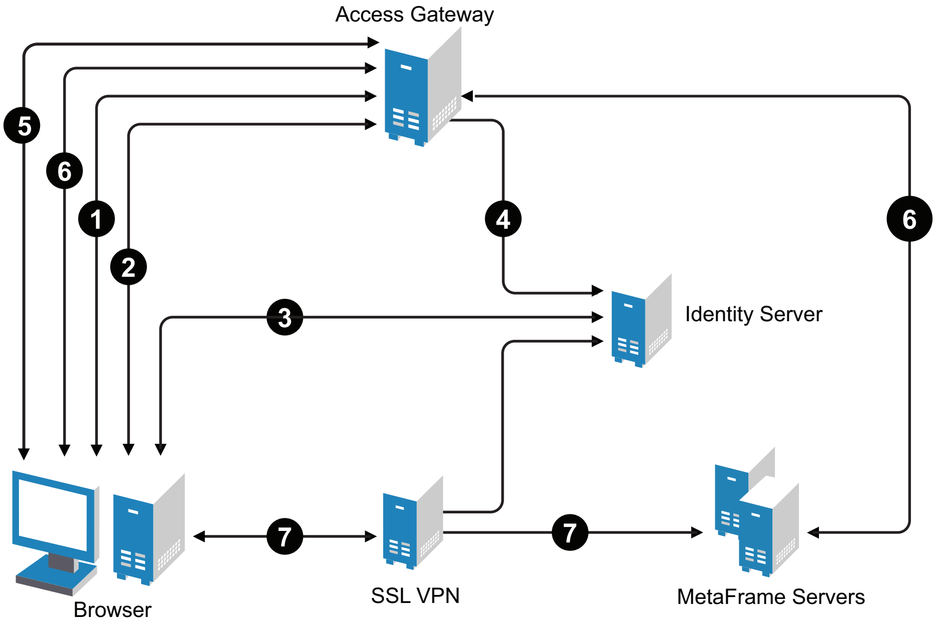
   ```
   /etc/init.d/novell-jcc restart
   ```

This imports the SSL VPN server to the new Administration Console. If you had configured multiple private IP addresses for the SSL VPN server, you can reconfigure them in the new Administration Console.

22 Configuring SSL VPN for Citrix Clients

The Access Manager can be configured to provide single sign-on for Citrix* clients. Figure 22-1 illustrates this process for the Citrix Web client.

Figure 22-1 *Citrix Client Configuration*



1. The client specifies the public DNS name of the Access Gateway that accelerates the Web Interface login page of the Citrix MetaFrame Presentation Server.
2. The Access Gateway redirects the user to the Identity Server for authentication, because the URL is configured as a protected resource.
3. The Identity Server authenticates the user's identity.
4. The Identity Server propagates the session information to the Access Gateway through the Embedded Service Provider.
5. The following activities take place:
   a. The Access Gateway has been configured with a Form Fill policy, which invokes the SSL VPN servlet along with the corresponding policy information for that user. The SSL VPN servlet creates a secure tunnel between the client and the SSL VPN server.
   b. On successful SSL VPN connection, the Access Gateway performs a single sign-on to the Citrix MetaFrame Presentation Server. The user is authenticated to both the Citrix Presentation Server and to the SSL VPN server.
6. The Web session containing the list of published applications in the Citrix Presentation server is served to the client through the Access Gateway.

7. When the user connects to the published application, the data goes through the secure tunnel that is formed between the client and the SSL VPN server.

22.1 Prerequisites

- ☐ NFuse server
- ☐ MetaFrame server
- ☐ Identity Server
- ☐ Access Gateway
- ☐ SSL VPN configured to use the same Identity Server as the Access Gateway.
- ☐ MetaFrame server must be placed in the protected network. The SSL VPN server must use its private network interface adapter to communicate with the network interface of the MetaFrame server.

22.2 Configuring the Access Gateway for Citrix Clients

1 Create a protected resource for the Citrix login page.

1a In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.

The reverse proxy can be set up to require SSL or not.

1b Click *Name of Proxy Service* > *Protected Resources* > *New*.

1c When configuring the protected resource, set up the following:

- Select a contract that requires authentication. Usually this is a Name/Password contract, but it can be a certificate contract if your NFuse server is configured to use certificates.
- For the URL Path List, specify the URL to the Citrix login page. This URL should include the filename of this login page.

For more information, see Section 13.4, “Configuring Protected Resources,” on page 207.

2 Create a Form Fill policy and assign it to the protected resource for the Citrix login page.

2a Click *Form Fill* > *Manage Policies* > *New*.

2b Name the Citrix policy, select *Access Gateway: Form Fill* as the type, then click *OK*.

2c In the *Actions* section, click *New* > *Form Fill*.

2d In the *Form Selection* section, identity the form on the Citrix login page.

2e In the *Fill Options* section, create the following:

- Username input field
- Password input field
- (Optional). If your login page requires a domain, add a domain input field.

2f In the *Submit Options* section, configure the following:

- Select *Auto Submit*.
- Select *Enable JavaScript Handling*.

- Click *Statements to Execute on Post*. Copy the Citrix Script found in the *Additional Resources* (http://www.novell.com/documentation/novellaccessmanager/index.html) section in the Novell Documentation site.

  In the script:

  Replace <*ag-url*> with the hostname of the Access Gateway that is accelerating the the SSL VPN server.

  Change the protocol to HTTPS if the secure protocol is used.

  If you want to use the custom login method, change the URL to:

  http://<ag-url>/sslvpn/custom-login

2g Configure any other options to match your form and your network.

For more information, see Section 30.3.2, “Creating a Form Fill Policy,” on page 492.

2h In the *Actions* section, click *New* > *Form Login Failure*.

Specify the procedures you want followed when login fails. For more information, see Section 30.3.3, “Creating a Login Failure Policy,” on page 497.

Citrix displays login failures via the query string, so you’ll need to use CGI matching.

2i Click *OK*, then click *Apply Changes*.

3 Click *Close*.

You should return to the Form Fill page for the protected resource.

4 Select the policy you just created, then click *Enable*.

5 Click *Configuration Panel*, then click *OK*.

6 On the Server Configuration page, click *OK*, then click *Update*.

Security and Certificate Management

V

This section discusses the following topics:

- Chapter 23, “Understanding How Access Manager Uses Certificates,” on page 355
- Chapter 24, “Managing Certificates,” on page 357
- Chapter 25, “Assigning Certificates to Access Manager Devices,” on page 375

Understanding How Access Manager Uses Certificates

23

Access Manager allows you to manage centrally stored certificates used for digital signatures and data encryption. eDirectory™ resides on the Administration Console is the main certificate store for all of the Access Manager components. If you use Novell® Certificate Server™, you can continue to create certificates there and import them into Access Manager.

By default, all Access Manager components (Identity Server, Access Gateway, SSL VPN, and J2EE agents) trust the local Access Manager CA. However, if the Identity Server is configured to use an SSL certificate signed externally, the trust store of the embedded service provider for each component must be configured to trust this new CA.

Certificate management commands issued from a secondary Administration Console can work only if the primary console is also running properly. Other commands can work independent of the primary console.

You can create and distribute certificates to the following components:

- **Identity Server:** Certificates allow you to provide secure authentication to the Identity Server and enable encrypted content from the Identity Server portal, via HTTPS. They also provide secure communications between trusted Identity Servers and user stores.

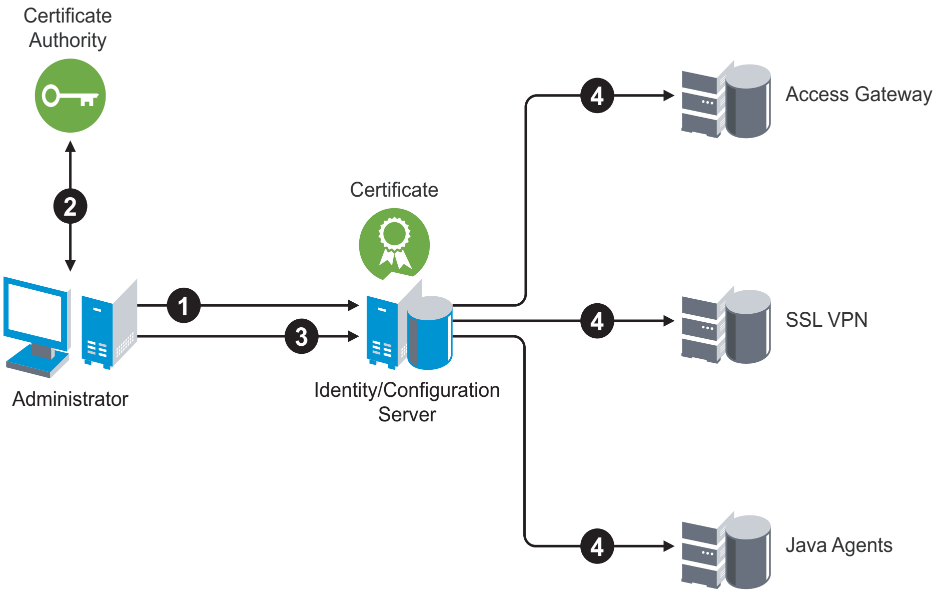
  Liberty and SAML 2.0 protocol messages that are exchanged between identity and service providers often need to be digitally signed. The Identity Server uses the signing certificate included with the metadata of a trusted provider to validate signed messages from the trusted provider. For protocol messages to be exchanged between providers using SSL, each provider must trust the CA of the other provider. You must import the CA used by the other provider.
- **Access Gateway:** Access Gateway uses server certificates and trusted roots to protect Web servers, provide single sign-on, and enable the product's data confidentiality features, such as encryption.
- **SSL VPN:** SSL VPN uses server certificates and trusted roots to secure access to non-HTTP applications.
- **J2EE Agent:** The J2EE agent uses certificates to establish trust between the J2EE Agent and the Identity Server and for SSL between the J2EE server and the Identity Server.

To ensure the validity of X.509 certificates, Access Manager supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) methods of verification.

Process Flow

You can install and distribute certificates to the Access Manager product components and configure how the components use certificates. This includes central storage, distribution, and expired certificate renewal. Figure 23-1 illustrates the primary administrative actions for certificate management in Access Manager:

Figure 23-1 *Certificate Management*



1. Create the certificate and generate a certificate signing request (CSR). See Section 24.1, "Creating Certificates," on page 357.
2. Send the CSR to the external CA for signing.

   A CA is a third-party or network authority that issues and manages security credentials and public keys for message encryption. The CA's certificate is held in the configuration store of the computers that trust the CA.
3. Import the signed certificate and CA chain into the configuration store. See Section 24.5, "Importing Public Key Certificates (Trusted Roots)," on page 368.
4. Assign certificates to devices. See Chapter 25, "Assigning Certificates to Access Manager Devices," on page 375.

If you are unfamiliar with public key cryptography concepts, see "Public Key Cryptography Basics" (http://www.novell.com/documentation/crt311/crtadmin/data/a2uqrry.html#a2uqrry) in the *Novell Certificate Server 3.1.1* Guide (http://www.novell.com/documentation/crt311/treetitl.html).

See Appendix C, "Certificates Terminology," on page 705 for information about certificate terminology.

Managing Certificates

24

Access Manager comes with certificates for testing purposes. The test certificates are called test-signing, test-encryption, test-provider, test-consumer, and test-connector. At a minimum you must create two SSL certificates: one for Identity Server test-connector and one for the Access Gateway reverse proxy. Then you replace the predefined certificates with the new ones.

If you install a secondary Administration Console, the certificate authority (CA) is installed with the first instance of eDirectory™, and the secondary consoles have eDirectory replicas, and therefore no CA software. All certificate management must be done from the primary Administration Console. Certificate management commands issued from a secondary Administration Console can work only if the primary console is also running properly. Other commands can work independent of the primary console.

IMPORTANT: Before generating any certificates with the Administration Console CA, make sure time is synchronized within one minute among all of your Access Manager devices. If the time of the Administration Console has a time that is before the device for which you are creating the certificate, the device rejects the certificate.

The following sections contain detailed information about creating and managing certificates for Access Manager:

- Section 24.1, “Creating Certificates,” on page 357
- Section 24.2, “Auto-Importing Certificates from Servers,” on page 367
- Section 24.3, “Importing a Private/Public Key Pair,” on page 367
- Section 24.4, “Exporting a Private/Public Key Pair,” on page 367
- Section 24.5, “Importing Public Key Certificates (Trusted Roots),” on page 368
- Section 24.6, “Renewing a Certificate,” on page 368
- Section 24.7, “Exporting a Public Certificate,” on page 370
- Section 24.8, “Enabling 4096k Keys,” on page 370
- Section 24.9, “Viewing Certificate Details,” on page 373

24.1 Creating Certificates

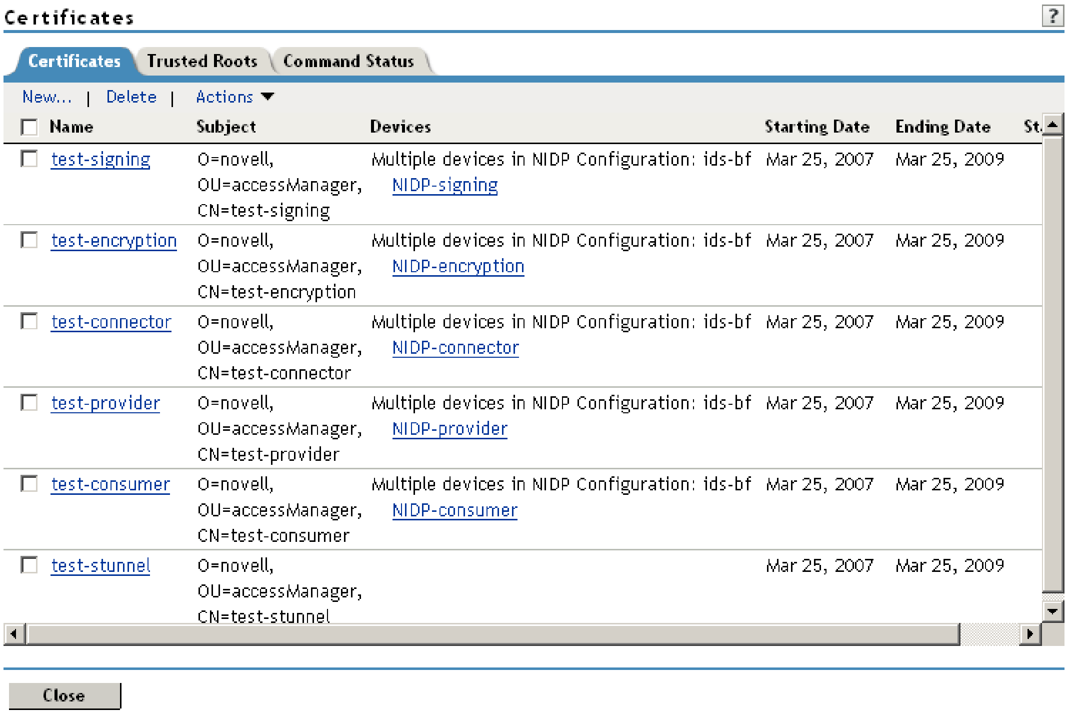
This task involves creating a certificate to be signed locally, or creating one that generates the CSR to be signed externally, which you later import after signing.

- Section 24.1.1, “Creating a Locally Signed Certificate,” on page 358
- Section 24.1.2, “Generating a Certificate Signing Request,” on page 365
- Section 24.1.3, “Importing a Signed Certificate,” on page 366

24.1.1 Creating a Locally Signed Certificate

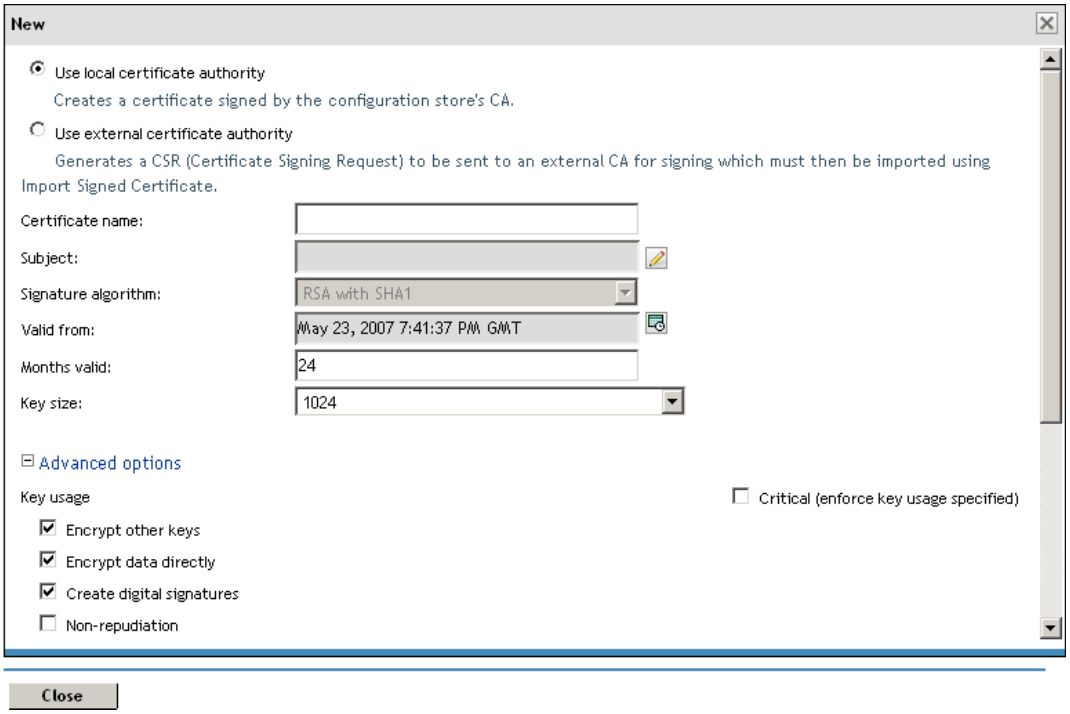
By default, the Access Manager installation process creates the local CA for you. eDirectory contains a CA that can issue and sign certificates, and a certificate server that generates or imports certificates and keys, and generate CSRs

1 In the Administration Console, click *Certificates*.



☐ Name	Subject	Devices	Starting Date	Ending Date	St.
☐ test-signing	O=novell, OU=accessManager, CN=test-signing	Multiple devices in NIDP Configuration: ids-bf NIDP-signing	Mar 25, 2007	Mar 25, 2009	
☐ test-encryption	O=novell, OU=accessManager, CN=test-encryption	Multiple devices in NIDP Configuration: ids-bf NIDP-encryption	Mar 25, 2007	Mar 25, 2009	
☐ test-connector	O=novell, OU=accessManager, CN=test-connector	Multiple devices in NIDP Configuration: ids-bf NIDP-connector	Mar 25, 2007	Mar 25, 2009	
☐ test-provider	O=novell, OU=accessManager, CN=test-provider	Multiple devices in NIDP Configuration: ids-bf NIDP-provider	Mar 25, 2007	Mar 25, 2009	
☐ test-consumer	O=novell, OU=accessManager, CN=test-consumer	Multiple devices in NIDP Configuration: ids-bf NIDP-consumer	Mar 25, 2007	Mar 25, 2009	
☐ test-stunnel	O=novell, OU=accessManager, CN=test-stunnel		Mar 25, 2007	Mar 25, 2009	

2 Click *New*.


New

● Use local certificate authority

Creates a certificate signed by the configuration store's CA.

○ Use external certificate authority

Generates a CSR (Certificate Signing Request) to be sent to an external CA for signing which must then be imported using Import Signed Certificate.

Certificate name:

Subject:

Signature algorithm: RSA with SHA1

Valid from: May 23, 2007 7:41:37 PM GMT

Months valid: 24

Key size: 1024

⊟ Advanced options

Key usage

☐ Critical (enforce key usage specified)

☑ Encrypt other keys

☑ Encrypt data directly

☑ Create digital signatures

☐ Non-repudiation

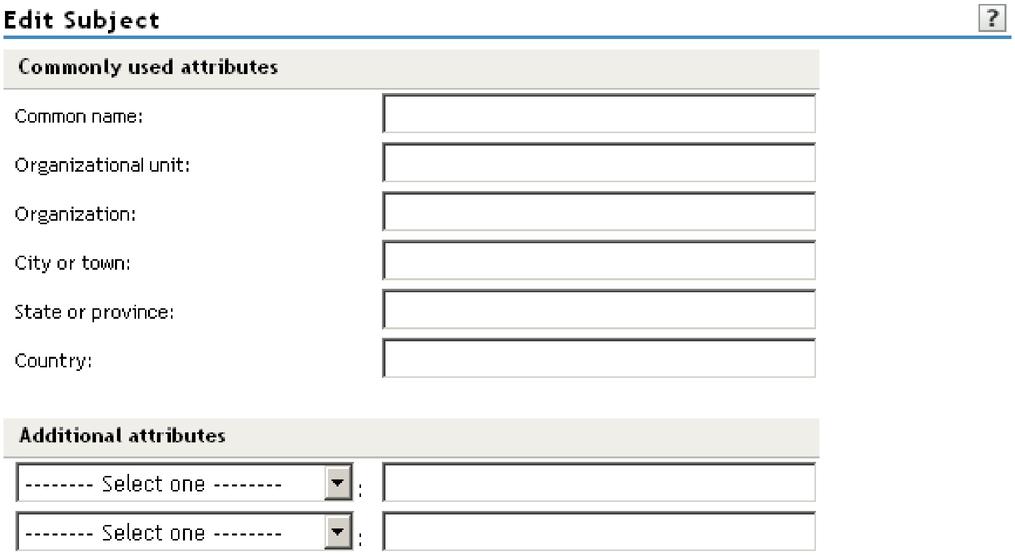
3 Select the following option:

Use local certificate authority: Creates a certificate signed by the local CA (or Organizational CA), and creates the private key. For information about creating a CSR, see Section 24.1.2, “Generating a Certificate Signing Request,” on page 365.

4 Fill in the following fields:

Certificate name: The name of the certificate. Pick a unique, system-wide name for the certificate that you can easily associate with the certificate’s purpose. The name must contain only alphanumeric characters and no spaces.

4a For *Subject*, click the *Edit* button to display a dialog box that lets you add the appropriate locality information types for the subject name.


Edit Subject

Commonly used attributes

Common name:

Organizational unit:

Organization:

City or town:

State or province:

Country:

Additional attributes

The subject is an X.500 formatted distinguished name that identifies the entity that is bound to the public key in an X.509 certificate. Choose the subject name that the browser expects to find in the certificate. The name you enter must be fully distinguished. Completing all the fields creates a fully distinguished name that includes the appropriate types (such as C for country, ST for state, L for location, O for organization, OU for organizational unit, and CN for common name). For example, cn=AcmeWebServer.ou=Sales.o=Acme.c=US.

The following attributes are the most common ones used in certificate subjects:

Common name: The name or IP of the Web server.

Enter just the value, for example AcmeWebServer. Do not include the type (cn=). The UI adds that for you.

For the Identity Server, this is the domain name of the base URL of the Identity Server configuration. This value cannot be an IP address or begin with a number, in order to ensure that trust does not fail between providers

Organizational unit: Describes departments or divisions.

Organization: Differentiates between organizational divisions.

City or town: Commonly referred to as the Locality.

State or province: Commonly referred to as the State.

Country: The country, such as US.

Use the *Additional Attributes* drop-down menus to add additional attributes. For more information about these attributes, see "Additional Attributes" on page 363.

5. Click *OK*, then fill in the following fields:

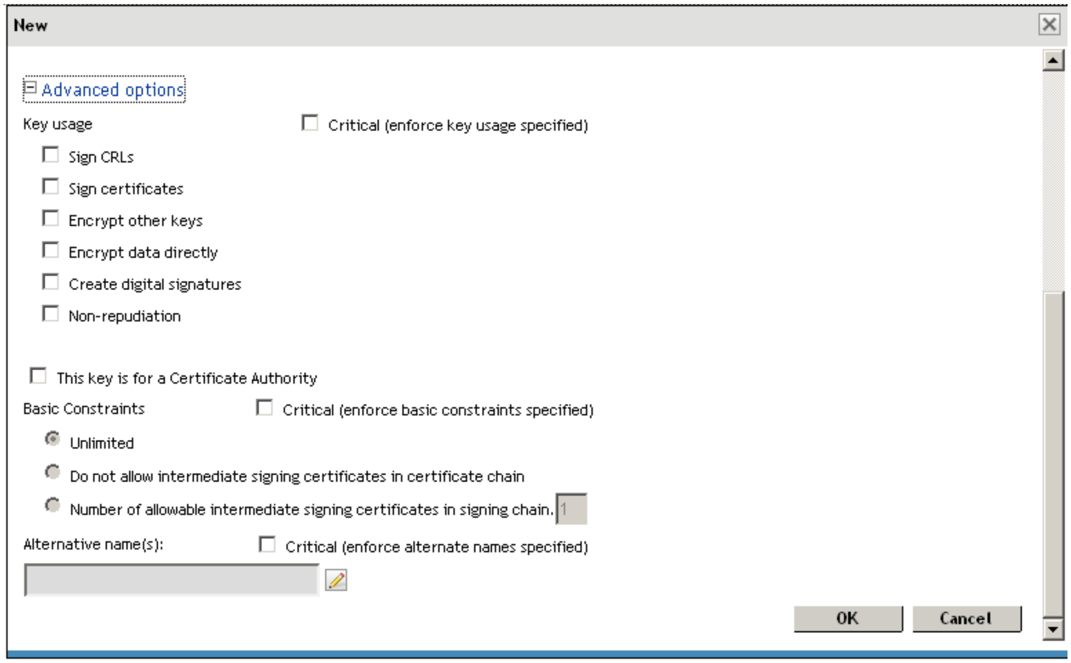
   Signature algorithm: The algorithm you want to use (SHA-1, MD-2, or MD-5). SHA-1 is currently recommended.

   Valid from: The date from which the certificate is valid. For externally signed certificates, the external certificate authority sets the validity period.

   Months valid: The number of months that the certificate is valid.

   Key size: The size of the key. Select 512, 1024, or 2048. 2048 bit is recommended. For 4096 key information, see Section 24.8, "Enabling 4096k Keys," on page 370.

6. (Optional) To configure advanced options, click *Advanced Options*.


New
⊟ Advanced options
Key usage
☐ Critical (enforce key usage specified)
☐ Sign CRLs
☐ Sign certificates
☐ Encrypt other keys
☐ Encrypt data directly
☐ Create digital signatures
☐ Non-repudiation
☐ This key is for a Certificate Authority
Basic Constraints
☐ Critical (enforce basic constraints specified)
● Unlimited
○ Do not allow intermediate signing certificates in certificate chain
○ Number of allowable intermediate signing certificates in signing chain. 1
Alternative name(s):
☐ Critical (enforce alternate names specified)
OK
Cancel

7 Configure the following options as necessary for your organization:

Critical: Specifies that an application should reject the certificate if the application does not understand the key usage extensions.

Sign CRLs: Specifies whether the certificate is used to sign CRLs (Certificate Revocation Lists).

Sign certificates: Specifies that the certificate is used to sign other certificates.

Encrypt other keys: Specifies that the certificate is used to encrypt keys.

Encrypt data directly: Encrypts data for private transmission to the key pair owner. Only the intended receiver can read the data.

Create digital signatures: Specifies that the certificate is used to create digital signatures.

Non-repudiation: Links a digital signature to the signer and the data. This prevents others from duplicating the signature because no one else has the signer's private key. Additionally, the signer cannot deny having signed the data.

This key is for a Certificate Authority: Specifies that this certificate is for the local configuration (eDirectory) certificate authority.

If you create a new CA, all the keys signed by the CA being replaced no longer have a trusted CA. Thus, you might also need to reassign the new CA to all the trust stores that contained the old CA.

8 Under *Basic Constraints*, configure the following options as necessary:

Critical: Enforces the basic constraints you specify.

Unlimited: Specifies no restriction on the number of subordinate certificates that the CA can verify.

Do not allow intermediate signing certificates in certificate chain: Prevents the CA from creating other CAs, but it can create server or user certificates.

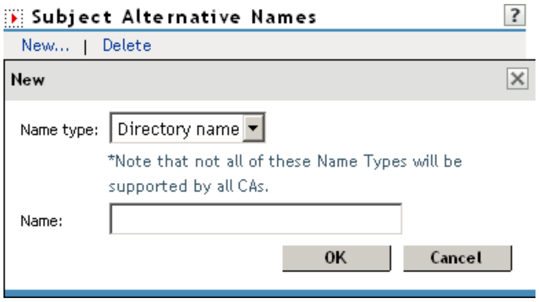
Number of allowable intermediate signing certificates in signing chain: Specifies how many subordinate certificates are allowed in the certificate chain. Values must be 1 or more. Entering 0 creates only entity objects.

Critical: Specifies that if an application does not understand the alternate name extensions, it should reject the certificate.

9 (Optional) To create subject alternative names used by the certificate, click the *Edit Subject Alternate Names* button.

Alternate names can represent the entity identified by the certificate. The certificate can identify the subject CN=www.OU=novell.O=com, but the subject can also be known by an IP address, such as 222.111.100.101, or a URI, such as www.novell.com, for example.

10 Click *New*.


Subject Alternative Names
?
New... | Delete
New
Name type:
Directory name
*Note that not all of these Name Types will be supported by all CAs.
Name:
OK
Cancel

Name Type: Names as specified by RFC 2459. Use the drop-down list to specify a name type, such as:

- **Directory name:** An X.500 directory name. The required format for the name is `.<attribute name>=<attribute value>`. For example:

  `.O=novell.C=US`

  Access Manager supports the following attributes:

  Country (C)
  Organization (O)
  Organizational Unit (OU)
  State or Province (S or ST)
  Locality (L)
  Common Name (CN)
- **IP Address:** An IP address such as 222.123.123.123
- **URI:** A URI such as www.novell.com.
- **Registered ID:** An ASN.1 object identifier.
- **DNS Name:** A domain name such as novell.com.
- **RFC822 Name:** An e-mail address.
- **X400 Name:** The messaging and e-mail standard specified by the ITU-TS (International Telecommunications Union - Telecommunication Standard Sector). It is an alternative to the more prevalent Simple Mail Transfer Protocol (SMTP) e-mail protocol. X.400 is common in Europe and Canada.
- **EDI Party:** EDI (Electronic Data Interchange) is a standard format for exchanging business data.

   - **Other:** A user-defined name.

   Name: The display alternative name.

11 Click *OK*.

See Also

- Section 24.3, “Importing a Private/Public Key Pair,” on page 367
- Section 24.4, “Exporting a Private/Public Key Pair,” on page 367
- Section 24.5, “Importing Public Key Certificates (Trusted Roots),” on page 368

Additional Attributes

Use the drop-down menus to add additional attributes. These values allow you to specify additional fields that are supported by eDirectory, and you can include them as part of the subject to further identify the entity represented by the certificate.

CN: The *Common name* attribute in the list of *Commonly used attributes* (OID: 2.5.4.3)

C: The *Country attribute* in the list of *Commonly used attributes* (OID: 2.5.4.6)

SN: The surname attribute (OID: 2.5.4.4)

L: The locality attribute, which is the *City or town* attribute in the list of *Commonly used attributes* (OID: 2.5.4.7)

ST: The *State or province* attribute in the list of *Commonly used attributes* (OID: 2.5.4.8)

S: The *State or province* attribute in the list of *Commonly used attributes* (OID: 2.5.4.8)

O: The Organization attribute in the list of Commonly used attributes (OID: 2.5.4.10)

OU: The Organizational unit attribute in the list of Commonly used attributes (OID: 2.5.4.11)

street: Text that the describes the street address (OID: 2.5.4.9)

serialNumber: Text that specifies the serial number of a device (OID: 2.5.4.5)

title: Text that describes the position or function of an object (OID: 2.5.4.12)

description: Text that describes the associated object (OID: 2.5.4.13)

searchGuide: Specifies a search filter (OID: 2.5.4.14)

businessCategory: Text that describes the kind of business performed by an organization (OID: 2.5.4.15)

postalAddress: Specifies address information required for the physical delivery of postal messages (OID: 2.5.4.16)

postalCode: Text that specifies the postal code of an object (OID: 2.5.4.17)

postOfficeBox: Text that specifies the post office box for the physical delivery of mail (OID: 2.5.4.18)

physicalDeliveryOfficeName: Text that specifies the name of the city or place where a physical delivery office is located (OID: 2.5.4.19)

telephoneNumber: Specifies a telephone number (OID: 2.5.4.20)

telexNumber: Specifies a telex number (OID: 2.5.4.21)

teletexTerminalIdentifier: Specifies an identifier for a telex terminal (OID: 2.5.4.22)

facsimileTelephoneNumber: Specifies the telephone number for a facsimile terminal (OID: 2.5.4.23)

x121Address: Specifies the address used in electronic data exchange (OID: 2.5.4.24)

internationalISDNNumber: Specifies an international ISDN number used in voice, video, and data transmission (OID: 2.5.4.25)

registeredAddress: Text that specifies the postal address for the delivery of telegrams or expedited documents (OID: 2.5.4.26)

destinationIndicator: Specifies an attribute used in telegram services (OID: 2.5.4.27)

preferredDeliveryMethod: Specifies the preferred delivery method for a message (OID: 2.5.4.28)

presentationAddress: Specifies an OSI presentation layer address (OID: 2.5.4.29)

supportedApplicationContext: Text that specifies the identifiers for the OSI application contexts in the application layer (OID: 2.5.4.30)

member: Specifies the distinguished name of an object associated with a group or a list (OID: 2.5.4.31)

owner: Text that specifies the name of an object that has responsibility for another object (OID: 2.5.4.32)

roleOccupant: Specifies the distinguished name of an object that fulfills an organizational role (OID: 2.5.4.33)

seeAlso: Specifies the distinguished name of an object that contains additional information about the same real world object (OID: 2.5.4.34)

userPassword: Specifies the object's password (OID: 2.5.4.35)

name: Text that specifies a name that is in the UTF-8 form of the ISO 10646 character set (OID: 2.5.4.41)

givenName: Text that specifies the given, or first name of an object (OID: 2.5.4.42)

initials: Text that specifies the initials of an object (OID: 2.5.4.43)

generationQualifier: Text that specifies the generation of an object, which is usually a suffix (OID: 2.5.4.44)

x500UniqueIdentifier: Specifies an identifier which distinguishes between objects when a DN has been reused (OID: 2.5.4.45)

dnQualifier: Specifies information which makes an object unique when information is being merged from multiple sources and objects could have the same RDNs (OID: 2.5.4.46)

enhancedSearchGuide: Specifies a search filter used by X.500 users (OID: 2.5.4.47)

protocolInformation: Specifies information which is used with the presentationAddress attribute (OID: 2.5.4.48)

distinguishedName: Specifies the distinguished name of an object (OID: 2.5.4.49)

uniqueMember: Specifies the distinguished name of an object associated with a group or a list (OID: 2.5.4.50)

houseIdentifier: Text that identifies a building within a location (OID: 2.5.4.51)

dmdName: Text that specifies a directory management domain (OID: 2.5.4.54)

E: Text that specifies an email address.

EM: Text that specifies an email address.

DC: Text that specifies the domain name for an object (OID: 0.9.2342.19200300.100.1.25)

uniqueID: Text that contains an RDN-type name that can be used to create a unique name in the tree (OID: 0.9.2342.19200300.100.1.1)

T: Text that specifies the name of the tree root object (OID: 2.16.840.1.113719.1.1.4.1.181)

OID: Text that specifies an object identifier in dot notation.

24.1.2 Generating a Certificate Signing Request

1. In the Administration Console, click *Certificates*, then click *New*.
2. Select the following option:

   Use external certificate authority: Generates a Certificate Signing Request (CSR) for you to send to the CA for signing. A third-party CA is managed by a third party outside of the eDirectory tree. An example of a third party CA is VeriSign*. After the signed certificate is received, you need to import the certificate. See Section 24.1.3, "Importing a Signed Certificate," on page 366.
3. Fill in the following fields:

   Certificate name: The name of the certificate. Pick a name unique, system-wide name for the certificate that you can easily associate with the certificate's purpose. The name must contain only alphanumeric characters and no spaces.

   Subject: An X.500 formatted distinguished name that identifies the entity that is bound to the public key in an X.509 certificate. Choose the subject name that the browser expects to find in the certificate. The name you enter must be fully distinguished. Completing all the fields creates a fully distinguished name that includes the appropriate types (such as C for country, ST for state, L for location, O for organization, OU for organizational unit, and CN for common name). For example, cn=AcmeWebServer.ou=Sales.o=Acme.c=US
4. Click the *Edit* button to display a dialog box that lets you add appropriate locality information types for the subject name.

   The following attributes are the most common ones used in certificate subjects:

   Common name: The name or IP of the Web server. Enter just the value. Do not enter the type (cn=). The UI adds it for you.

   Organizational unit: Describes departments or divisions.

   Organization: Differentiates between organizational divisions.

City or town: Commonly referred to as the Locality.

State or province: Commonly referred to as the State.

Country: The country, such as US.

Use the *Additional Attributes* drop-down lists to add additional attributes. These values allow you to specify additional fields that are supported by eDirectory, and you can include them as part of the subject to further identify the entity represented by the certificate.

5. Click *OK*, then fill in the following fields:

   Signature algorithm: The algorithm you want to use (SHA-1, MD-2, or MD-5). SHA-1 is currently recommended.

   Valid from: The date from which the certificate is valid. For externally signed certificates, the external certificate authority sets the validity period.

   Months valid: The number of months that the certificate is valid.

   Key size: The size of the key. Select 512, 1024, or 2048. 2048 bit is recommended. For 4096 key information, see Section 24.8, "Enabling 4096k Keys," on page 370.
6. If necessary, fill in the certificate fields, which are described in Section 24.1.1, "Creating a Locally Signed Certificate," on page 358.
7. Click *OK*.
8. On the Certificate Details page, copy the CSR data and send the information to the external CA.

   The certificate status is CSR Pending until you import the signed certificate.
9. Click *Close*.

Continue with Section 24.1.3, "Importing a Signed Certificate," on page 366 after you receive the signed certificate and the trusted root (CA chain).

24.1.3 Importing a Signed Certificate

After you receive the signed certificate and the CA chain, you must import it. There are several ways in which the CA can return the certificate. Typically, the CA either returns one or more files each containing one certificate, or returns a file with multiple certificates in it.

1. In the Administration Console, click *Certificates*, then click the certificate name.
2. Click *Import Signed Certificate*.
3. In the Import Signed Certificate dialog box, browse to locate the certificate data file, or paste the certificate data text into the *Certificate data text* field.
4. To import the CA chain, click *Add trusted root*, then locate the certificate data.
5. Click *Add intermediate certificate* if you need to continue adding certificates to the chain.
6. Click *OK*, then click *Close* on the Certificate Details page.

The certificate is now available for use by Access Manager devices.

If you receive an error when attempting to import the certificate, see Chapter 44, "Troubleshooting Certificate Issues," on page 691.

24.2 Auto-Importing Certificates from Servers

You can import certificates from other servers, such as an LDAP server, and make them available for use in Access Manager. You must provide the IP address, port, and certificate name.

1. In the Administration Console, click *Access Manager* > *Certificates* > *Trusted Roots* > *Auto-Import from Server*.
2. Fill in the following fields:

   Server IP Address: Specifies the server IP address. You can use a DNS name.

   Server Port: Specifies the server port.

   Certificate Name: Specifies a unique name of the certificate to store in Access Manager.
3. Click *OK*.

24.3 Importing a Private/Public Key Pair

If you created a key pair that was exported from another certificate management system, you can import the key pair and then assign it to an Access Manager device. The file needs to be in PKCS12 (*`.pfx`) or (*`.p12`) format.

1. In the Administration Console, click *Certificates*.
2. Choose *Actions* > *Import Private/Public Keypair*.
3. Fill in the following fields:

   Certificate name: The name of the certificate. This is a system-wide, unique name used by Access Manager.

   Password: Type the encryption/decryption password established when exporting the certificate.

   Certificate data file: The certificate file to import. You can browse to locate the `.pfx` or `.p12` file.

   Certificate data text: An editable field used to enter or paste certificate data text. This is valid if your PKCS12 file is in Base64-encoded format. The first line of the data is `-----BEGIN PKCS12-----`.

   Overwrite an existing certificate with the same name: Specifies whether to replace any existing certificates with the same name as this one.
4. Click *OK*.

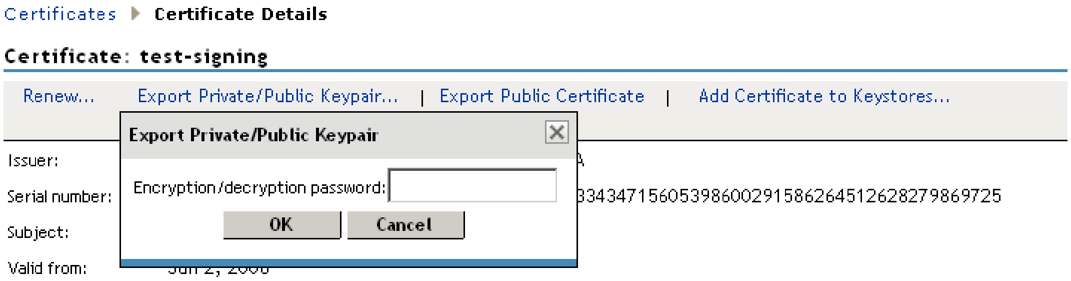
If you receive an error when importing the certificate, the error comes from either NICI or PKI. For a description of these error codes, see Novell® Certificate Server Error Codes and Novell International Cryptographic Infrastructure (http://www.novell.com/documentation/nwec/index.html). For general certificate import issues, see Section 44.2, “Importing an External Certificate Key Pair,” on page 692.

24.4 Exporting a Private/Public Key Pair

When you create a certificate, you can specify whether it is exportable. If a key is exportable, it can be extracted and put in a file along with the associated certificate. The file is written in an industry standard format, PKCS#12, which allows it to be transported to other platforms. It is encrypted with a user-specified password to protect the private key.You can export private certificates to obtain a backup copy of the key, to move the key to a different server, or to share the key between servers.

You cannot export a certificate if you created it for NetWare® Access Gateway and enabled the *Do not allow private key to be exportable option* while creating the certificate.

1. In the Administration Console, click *Certificates*.
2. On the Certificates page, click the certificate.
3. On the Certificate Details page, click *Export Private/Public Keypair*.


Certificates ▸ Certificate Details
Certificate: test-signing
Renew...
Export Private/Public Keypair...
Export Public Certificate
Add Certificate to Keystores...
Export Private/Public Keypair
Encryption/decryption password:
OK
Cancel
Issuer:
Serial number:
Subject:
Valid from:
33434715605398600291586264512628279869725

4. Specify the password in the *Encryption/decryption* password field, then click OK.

IMPORTANT: Remember this password because you need it to re-import the key.

24.5 Importing Public Key Certificates (Trusted Roots)

You import trusted roots so that the specific device can trust the certificate sent by other computers at runtime. After you import a trusted root, you can assign it to the proper trust store associated with a device, which allows the device to trust certificates signed by the trusted root.

1. In the Administration Console, click *Access Manager*, then click *Certificates*.
2. Click the *Trusted Roots* tab.
3. Click *Import,* then fill in the following fields:

   Certificate name: The name of the certificate. This is a system-wide, unique name used by Access Manager. If you are importing this certificate into a Proxy Trust Store used by the NetWare Access Gateway, the name must be from one to eight characters long. The other trust stores accept longer names.

   Certificate data file: The certificate file to import. You can browse to locate the file or copy and paste text into the *Certificate data text* field.

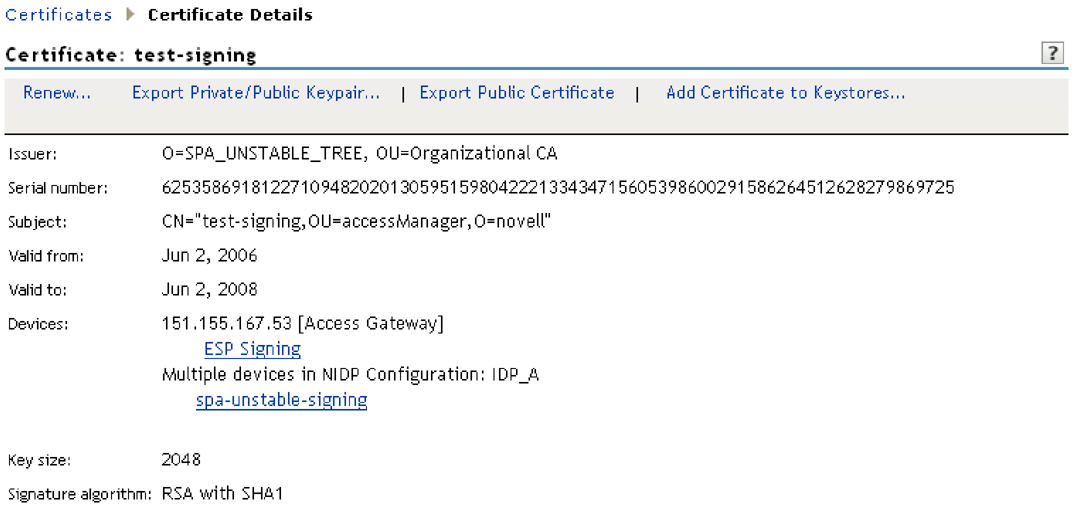
   Certificate data text: An editable field used to enter or paste Base64-encoded certificate data text.
4. Click *OK*.

24.6 Renewing a Certificate

The Certificate Details page lists the properties of a certificate, such as certificate type, name, subject, and assigned keystores. This page also includes the original CSR. If the certificate has expired, you can cut and paste its text to send it to the CA to get a renewed certificate, then import the newly signed certificate.

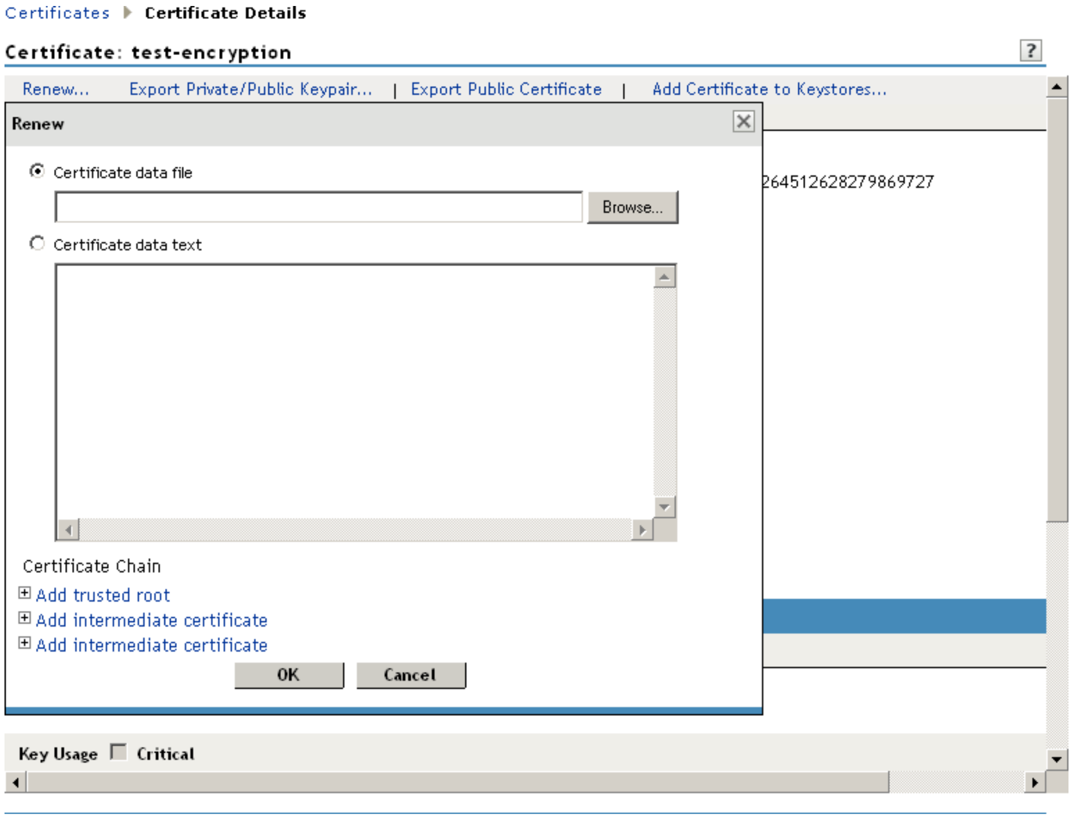
1 In the Administration Console, click *Certificates*.

2 Click the certificate name.


Certificates
Certificate Details
Certificate: test-signing
?
Renew...
Export Private/Public Keypair...
Export Public Certificate
Add Certificate to Keystores...

Issuer: O=SPA_UNSTABLE_TREE, OU=Organizational CA
Serial number: 62535869181227109482020130595159804222133434715605398600291586264512628279869725
Subject: CN="test-signing,OU=accessManager,O=novell"
Valid from: Jun 2, 2006
Valid to: Jun 2, 2008
Devices: 151.155.167.53 [Access Gateway]
ESP Signing
Multiple devices in NIDP Configuration: IDP_A
spa-unstable-signing
Key size: 2048
Signature algorithm: RSA with SHA1

3 Click *Renew*.


Certificates
Certificate Details
Certificate: test-encryption
?
Renew...
Export Private/Public Keypair...
Export Public Certificate
Add Certificate to Keystores...
Renew
Certificate data file
Browse...
Certificate data text
Certificate Chain
Add trusted root
Add intermediate certificate
Add intermediate certificate
OK
Cancel
Key Usage
Critical

● Certificate data file
○ Certificate data text
264512628279869727
☐ Critical

4 On the Renew page, browse to locate and select the certificate, then click *OK*.

24.7 Exporting a Public Certificate

You can export a trusted root or a public key certificate to a file so that a client can use it to verify the certificate chain sent by a cryptography-enabled application, or to have a backup copy of the file.

You can export the certificate in the following formats:

- DER-encoded (`.der`) to a file.
- PEM-encoded to a file. This is a Base64-encoded DER certificate that is enclosed between BEGIN CERTIFICATE and END CERTIFICATE tags.
- Base64-encoded to the system Clipboard. You can then pasted it directly into a cryptography-enabled application.

To export the public certificate:

1. In the Administration Console, click *Certificates*.
2. From either the *Certificates* tab or the *Trusted Roots* tab, click the certificate name.
3. On the Certificate Details page, click *Export Public Certificate*, then click the file type.
4. Save the output file to the location of your choosing.

24.8 Enabling 4096k Keys

The Bouncy Castle provider that comes with Access Manager can be used to handle greater key sizes. When enabling 4096k keys, ensure that you configure each component that uses the certificate. This key size is not available for the NetWare Access Gateway’s reverse proxies. However, if an identity provider uses 4k keys, the embedded service provider on the NetWare Access Gateway must be configured to trust the key.

The basic functionality for using cryptographic techniques in Java is provided by the Java Cryptography Architecture* (JCA) and Java Cryptography Extension* (JCE). This architecture is what is referred to as provider-based (pluggable) architecture. In this case, it means that the JCE and JCA provide a set of classes and interfaces that an application developer writes to, together with factories that enable the creation of the objects that conform to the interfaces and classes.

- Section 24.8.1, “Understanding Jurisdiction Policy Files,” on page 370
- Section 24.8.2, “Downloading JCE Files,” on page 371
- Section 24.8.3, “Copying the Files to the Components,” on page 371
- Section 24.8.4, “Enabling 4k Support Deployment,” on page 372
- Section 24.8.5, “Verifying 4k Key Support,” on page 372

24.8.1 Understanding Jurisdiction Policy Files

Because of various export and import restrictions in various geographies, the Java Development Kit (JDK*) download ships with a set of policy files that place certain restrictions on the key sizes that can be used. Key sizes are limited in general to 128 bits (except for the symmetric cipher Triple-DES), and RSA key generation is limited to 2,048 bits. The easiest way to deal with this restriction if it need not apply to you is to download the unrestricted policy files.

24.8.2 Downloading JCE Files

Navigate to the J2SE 1.4.2 Web site (http://java.sun.com/j2se/1.4.2/download.html) and locate *Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 1.4.2.* Click *Download* to access the `jce_policy-1_4_2.zip` file. The JCE files are the only files you need to download, because Access Manager comes with the Bouncy Castle JAR files.

You must install the policy files in the Java runtime that you are using; you need `root` access or the assistance of a `root` user to do so.

24.8.3 Copying the Files to the Components

Both Linux and NetWare Access Gateways include the Bouncy Castle provider `.jar` file installed in the embedded service provider. A standalone Administration Console server without an Identity Server can use the Bouncy Castle provider `.jar` file from an identity provider or embedded service provider.

Typically these files are installed (copied) in the `*JAVA_HOME*/jre/lib/security` directory, where JAVA_HOME is the home directory path of JVM used by Access Manager components. For example, `/opt/novell/java`.

IMPORTANT: Back up your Access Manager configuration, and copy the `.zip` file to a secure location on another machine (see Backup and Restore), and stop all Access Manager components.

- "Identity Provider" on page 371
- "Linux Access Gateway" on page 371
- "NetWare Access Gateway" on page 372

Identity Provider

1. Back up the following JCE (`.jre` jurisdiction policy) files:
    - `/usr/lib/java/jre/lib/security`
    - `/opt/novell/java/jre/lib/security`
2. Back up the following `.jre` (Bouncy Castle) files:
    - `/usr/lib/java/jre/lib/ext`
    - `/opt/novell/java/jre/lib/ext`
3. Copy the Bouncy Castle provider `.jar` file from

   /var/opt/novell/tomcat4/webapps/nidp/WEB-INF/lib/bcprov-jdk14-128.jar

   to

   `/opt/novell/j2sdk1.4.2_12/jre/lib/ext/bcprov-jdk14-128.jar`.
4. Copy the JCE jurisdiction policy JAR files to `/opt/novell/j2sdk1.4.2_12/jre/lib/security/`.

Linux Access Gateway

1. Back up the following JCE (`.jre` jurisdiction policy) file:
    - `/opt/novell/java/jre/lib/security`

2. Back up the following `.jre` (Bouncy Castle) files:
   - `/opt/novell/java/jre/lib/ext`
3. Copy the Bouncy Castle provider `.jar` file from

   `/var/opt/novell/tomcat4/webapps/nesp/WEB-INF/lib/bcprov-jdk14-128.jar`

   to

   `/opt/novell/java/jre/lib/ext/bcprov-jdk14-128.jar`.
4. Copy the JCE jurisdiction policy JAR files to `/opt/novell/java/jre/lib/security/`.

NetWare Access Gateway

1. Copy the Bouncy Castle provider `.jar` file from

   `sys:\tomcat\4\webapps\nesp\web-inf\lib\bcprov-jdk14-128.jar`

   to

   sys:\java\lib\ext\bcprov-jdk14-128.jar.
2. Copy the JCE jurisdiction policy JAR files to `sys:\java\lib\security\`.

24.8.4 Enabling 4k Support Deployment

1. Enable the provider by adding it to the `java.security` file in the *`JAVA_HOME`*`/jre/lib/security` directory.

   This file contains a section that lists JCA/JCE providers with their precedence (among other things). Add the Bouncy Castle provider second in the list, as follows:

   ```
   security.provider.1=sun.security.provider.Sun
   security.provider.2=org.bouncycastle.jce.provider.BouncyCastleProv
   ider
   security.provider.3=com.sun.net.ssl.internal.ssl.Provider
   security.provider.4=com.sun.rsajca.Provider
   security.provider.5=com.sun.crypto.provider.SunJCE
   ```

   The list might be different, depending on which version of Java you have installed. Ensure that the Bouncy Castle provider is in the second position as shown in the list above. This alters the cryptographic behavior of any process using this JVM. Do not put the Bouncy Castle provider at the top of the list, because some Java system software packages rely on the Sun providers being the first ones in the list, and processing can stop working if they aren't positioned correctly.
2. Restart all Access Manager components.

24.8.5 Verifying 4k Key Support

If your Identity Server is on the same machine as the Administration Console, you can verify 4k key support by creating a temporary certificate minted with the local CA that is 4k.

1. Log in to Access Manager.

2. Click *Access Manager* > *Certificates*.
3. Click *New* to create a certificate, and leave the *Local certificate authority* option enabled.
4. Specify a valid name for the test certificate, such as test_4k.
5. Change the *Key size* to 4096.
6. Click *OK*.
7. The system prompts you for a subject name, such as 4ktest.sso.novell.com.

   You only need to specify a common name, and can leave other fields blank.
8. Click *OK* twice.

You can see the certificate in the list, if you click on it, the details are displayed. The key size should be 4096.

To verify 4k key support on an Access Gateway:

1. Export a public key certificate that has a 4k key size (such as the test certificate created to verify the Administration Console's 4k key support described above.)
2. Import the public key certificate into the *Trusted Roots* tab.
3. Add the public key certificate into the embedded service provider or identity provider trust store.
4. Use keytool to view the actual trust store on the embedded service provider or identity provider to determine if the 4k public key certificate was added to the keystore.

   The embedded service provider keytool command for the Linux Access Gateway is:

   ```
   /opt/novell/java/jre/bin/keytool -v -list -keystore /chroot/
   lag/opt/novell/devman/jcc/certs/esp/truststore.keystore
   ```

   The identity provider keytool command is:

   ```
   /opt/novell/j2sdk1.4.2_12/bin/keytool -v -list -keystore /opt/
   novell/devman/jcc/certs/idp/truststore.keystore
   ```

24.9 Viewing Certificate Details

The Certificate Details page lists the properties of a certificate, such as certificate type, name, subject, and assigned keystores. The fields are not editable.

1. In the Administration Console, click *Access Manager* > *Certificates*.
2. Click the name of a certificate.

   The Certificate Details page contains the following information about the certificate:

   Issuer: Displays the name of the CA that created the certificate.

   Serial number: Displays the serial number of the certificate.

   Subject: Displays the subject name of the certificate.

   Valid from: Displays the first date and time that the certificate is valid.

   Valid to: Displays the date and time that the certificate expires.

   Devices: Indicates the devices that are configured to hold this certificate on their file system.

   Key size: Indicates the key size that was used to create the certificate.

   Signature algorithm: Indicates the signature algorithm that was used to create the certificate.

Finger print (MD5): Displays the certificate's message digest that was calculated with the MD5 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, a user can verify that a certificate is the one they think it is by matching this published MD5 fingerprint with the MD5 fingerprint on the local certificate.

Finger print (SHA1): Displays the certificate's message digest that was calculated with the SHA1 algorithm. It is embedded into the certificate at creation time. It can be used to uniquely identify a certificate. For example, a user can verify that a certificate is the one they think it is by matching a published SHA1 fingerprint with the SHA1 fingerprint on the local certificate.

Subject Alternate Names: Critical: Indicates whether an application should reject the certificate if the application does not understand the alternate name extensions. Any configured alternate names are displayed in the list.

Key Usage: Critical: Indicates whether an application should reject the certificate if the application does not understand the key usage extensions.

Sign CRLs: Indicates whether the certificate is used to sign CRLs (Certificate Revocation Lists).

Sign certificates: Indicates that the certificate is used to sign other certificates.

Encrypt other keys: Indicates that the certificate is used to encrypt keys.

Encrypt data directly: Indicates that the certificate encrypts data for private transmission to the key pair owner. Only the intended receiver can read the data.

Create digital signatures: Indicates that the certificate is used to create digital signatures.

Non-repudiation: Indicates that the certificate links a digital signature to the signer and the data. This prevents others from duplicating the signature because no one else has the signer's private key. Additionally, the signer cannot deny having signed the data.

CRL Distribution Points: Displays a list of Certificate Revocation List (CRL) distribution points that are embedded into the certificate as an extension at certificate creation time. Implementations search the CRL from each distribution point (the distribution point is usually a URI that points to a store of revoked certificates) to see whether a certificate has been revoked.

Authority Info Access (OCSP): Displays a list of Online Certificate Status Protocol (OCSP) responders that are embedded into the certificate as an extension at certificate creation time. Implementations query the OCSP responder to see whether a certificate has been revoked.

Assigning Certificates to Access Manager Devices

25

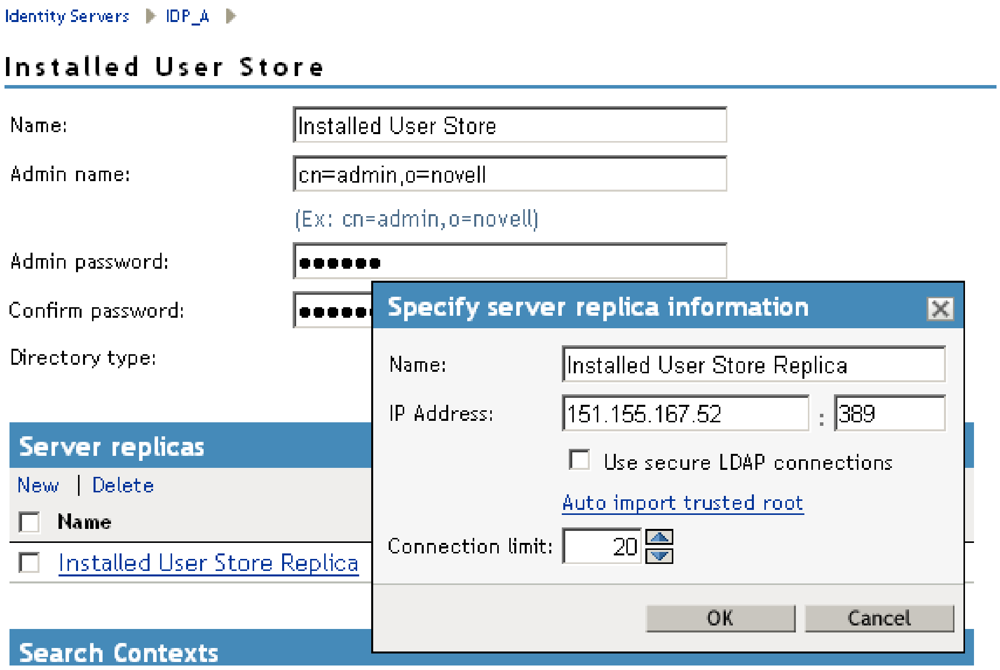
After you assign certificates to devices, the certificates are placed in keystores. Ensure that you update the device so that the certificates are pushed into active use.

This section discusses how you update, renew, and assign certificates to Access Manager devices.

25.1 Importing a Trusted Root to the LDAP User Store

When you specify the settings of a user store for an Identity Server configuration, or add a user store, you can import the trusted root certificate to the LDAP user store device.

1. In the Administration Console, click *Identity Servers > Edit > Local > [User Store]*.
2. Under *Server Replicas*, click the name of the server replica.



Installed User Store

Name: Installed User Store

Admin name: cn=admin,o=novell

(Ex: cn=admin,o=novell)

Admin password: ●●●●●●

Confirm password: ●●●●●●

Directory type:

Specify server replica information

Name: Installed User Store Replica

IP Address: 151.155.167.52 : 389

☐ Use secure LDAP connections

Connection limit: 20

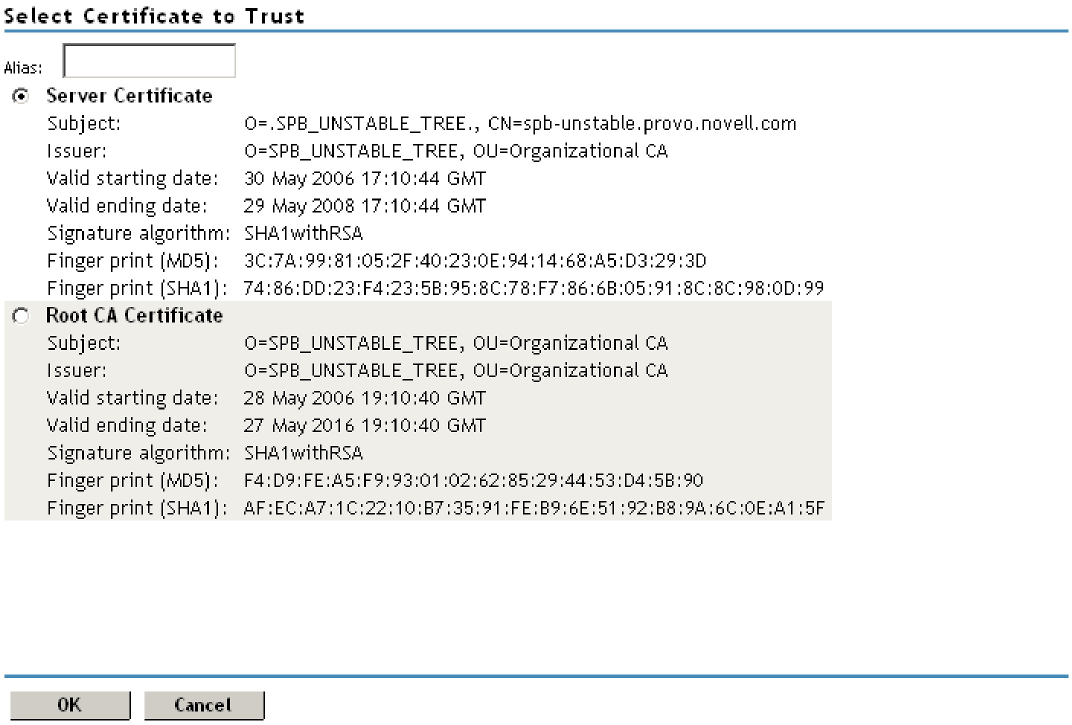
3 Enable the *Use secure LDAP connections* option.

This option allows SSL communication to occur between the Identity Server and the user store.

4 Click *Auto import trusted root.*

5 Click *OK* to confirm the import.

Ensure that you have pop-ups enabled, or the browser cannot display the Confirm dialog box.


Select Certificate to Trust

Alias:

● Server Certificate

Subject: O=.SPB_UNSTABLE_TREE., CN=spb-unstable.provo.novell.com

Issuer: O=SPB_UNSTABLE_TREE, OU=Organizational CA

Valid starting date: 30 May 2006 17:10:44 GMT

Valid ending date: 29 May 2008 17:10:44 GMT

Signature algorithm: SHA1withRSA

Finger print (MD5): 3C:7A:99:81:05:2F:40:23:0E:94:14:68:A5:D3:29:3D

Finger print (SHA1): 74:86:DD:23:F4:23:5B:95:8C:78:F7:86:6B:05:91:8C:8C:98:0D:99

○ Root CA Certificate

Subject: O=SPB_UNSTABLE_TREE, OU=Organizational CA

Issuer: O=SPB_UNSTABLE_TREE, OU=Organizational CA

Valid starting date: 28 May 2006 19:10:40 GMT

Valid ending date: 27 May 2016 19:10:40 GMT

Signature algorithm: SHA1withRSA

Finger print (MD5): F4:D9:FE:A5:F9:93:01:02:62:85:29:44:53:D4:5B:90

Finger print (SHA1): AF:EC:A7:1C:22:10:B7:35:91:FE:B9:6E:51:92:B8:9A:6C:0E:A1:5F

6. Select one of the certificates in the list.

   You are prompted to choose either a server certificate or a root CA certificate. To trust one certificate, choose *Server Certificate*. Choose *Root CA Certificate* to trust any certificate signed by that certificate authority.

7. Specify an alias, then click *OK*.

   You use the alias to identify the certificate in Access Manager.

8. On the User Store page, click *OK*.
9. Restart the Identity Server.

25.2 Replacing Identity Server SSL Certificates

This procedure allows you to replace a trusted root certificate that is stored in the trust store assigned to the Identity Server. You must create an SSL certificate for the Identity Server and then replace the predefined test-connector certificate that comes with Access Manager. You can also replace the test-provider and test-consumer certificates in the *NIDP-provider* and *NIDP-consumer* keystores. The steps for replacing the signing, encryption, provider, and consumer certificates are similar.

You can also add the trusted roots to the trust stores used by the Identity Server, or auto-import them from a server. The NIDP trust store is the certificate container for CA certificates associated with the Identity Server.

You can also access the OCSP trust store to add OCSP server certificates. Online Certificate Status Protocol is a method used for checking the revocation status of a certificate. For this feature, you must set up an OCSP server. The Identity Server sends an OCSP request to the OCSP server to determine if a certain certificate has been revoked. The OCSP server replies with the revocation status. If this revocation checking protocol is used, the Identity Server does not cache or store the information in the reply, but sends a request every time it needs to check the revocation status of a certificate. The OCSP reply is signed by the OCSP server. To verify that it was signed by the correct OCSP server, the OCSP server certificate needs to be added to this trust store. The OCSP server certificate itself is added to the trust store, not the CA certificate

1. In the Administration Console, click *Identity Servers* > *Edit* > *Security*.
2. Click the certificate link that you want to replace:

   Encryption: Displays the encryption certificate keystore. The encryption certificate is used to encrypt specific fields or data in the assertions.

   Signing: Displays the signing certificate keystore. Click this option to access the keystore and replace the signing certificate as necessary. The signing certificate is used to sign the assertion or specific parts of the assertion.

   SSL: Displays the SSL connector keystore. Click this option to access the keystore and replace the SSL certificate as necessary. This certificate is used for SSL connections.

   Provider: Displays the identity provider keystore. Click this option to access the keystore and replace the identity provider certificate.

   Consumer: Displays the identity consumer keystore. Click this option to access the keystore and replace the identity consumer certificate as necessary.

3. Click *Replace*.

   NOTE: A keystore stores only one certificate at a time. When you replace a certificate, you overwrite the existing one.

4. In the Replace dialog box, click the *Select Certificate* icon and browse to select the certificate you created in Section 24.1, “Creating Certificates,” on page 357.
5. Click *OK*.
6. Click *OK* in the Replace dialog box.
7. Restart Tomcat, as prompted by the system.

   The system restarts Tomcat for you if you click *Restart Now* at the prompt. If you want to restart at your convenience, select *Restart Later* and then manually restart Tomcat via ssh. Enter `/etc/init.d/novell-tomcat4 restart`, then press Enter.
8. Update the Identity Server configuration on the Servers page, as prompted.

25.3 Assigning Certificates to an Access Gateway

The Access Gateway can be configured to use certificates for SSL communication with three types of entities (see Section 14.5, “Managing Access Gateway Certificates,” on page 247):

- **Identity Server:** The Access Gateway uses the embedded service provider to communicate with the Identity Server. The Access Manager CA automatically generates the required certificates for secure communication when you set up a trusted relationship with the Identity Server. To manage these certificates in the Administration Console, click *Access Gateways > [Configuration Link] > Service Provider Certificates*. For more information, see Section 14.5.1, “Managing Embedded Service Provider Certificates,” on page 247.
- **Client browsers:** You can enable SSL communication between the client browsers and the Access Gateway. When setting up this feature, you can either have the Access Manager CA automatically generate a certificate key or you can select a certificate key you have already imported (or created) for the reverse proxy. To manage this certificate in the administration console, click *Access Gateways > [Configuration Link] > [Name of Reverse Proxy]*. For more information, see Section 13.1, “Creating a Reverse Proxy and Proxy Service,” on page 200.
- **Protected Web servers:** You can enable SSL communication between the Access Gateway and the Web servers it is protecting. This option is only available if you have enabled SSL communication between the browsers and the Access Gateway. You can enable SSL or mutual SSL. To manage these certificates in the Administration Console, click *Access Gateways > [Configuration Link] > [Name of Reverse Proxy] > [Name of Proxy Service] > Web Servers*. For more information, see Section 13.3, “Configuring the Web Servers of a Proxy Service,” on page 205.

25.4 Assigning Certificates to J2EE Agents

To enable the J2EE agent for SSL, you must set up the following trust relationships:

- The J2EE server with the Identity Server
- The J2EE agent with the Identity Server

For instructions on setting up these certificates, see “Configuring SSL Certificate Trust” in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*.

25.5 Configuring SSL for Authentication between the Identity Server and Access Gateway

By default, all Access Manager components (Identity Server, Access Gateway, SSL VPN, and J2EE agents) trust the certificates signed by the local CA. However, if the Identity Server is configured to use an SSL certificate signed externally, the trusted store of the service provider for each component must be configured to trust this new CA. Import the public certificate of the CA into the following trust stores:

- For an Access Gateway, click *Access Gateways > [Configuration Link] > Service Provider Certificates > Trusted Roots*.
- For a J2EE agent, click *J2EE Agents > Edit > Trusted Roots*.
- For an SSL VPN server, click *SSL VPNs > Edit > SSL VPN Certificates > Trusted Root*.

If an Access Gateway, a J2EE agent, or an SSL VPN server is configured to use an SSL certificate signed externally, the trusted store of the Identity Server must be configured to trust this new CA. Import the public certificate of the CA into the Identity Server configuration that the component is using for authentication.

In the Administration Console, click *Identity Servers > [Configuration Assignment] > Security > NIDP Trust Store* and add the certificate to the Trusted Roots list.

NOTE: Whenever you replace certificates on a device, you must update the Identity Server configuration (by clicking *Update Servers* on the Servers page), or restart the Access Gateway ESP application.

25.6 Changing a Non-Secure (HTTP) Environment to a Secure (HTTPS) Environment

If you are running in a non-secure staging environment, and you're ready to move to production, you must perform the following steps to enable security.

1. Change the Identity Server configuration protocol to HTTPS. (See "Configuring Secure Communication on the Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*)
2. Replace the test certificates with your own. (See "Enabling SSL Communication" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*)
3. Reimport metadata for trusted service and identity providers. (See Section 9.3, "Reimporting a Trusted Provider's Metadata," on page 144.)
4. Change the Access Gateway configuration to HTTPS. (See "Configuring the Access Gateway for SSL" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*)

25.7 Creating Keystores and Trust Stores

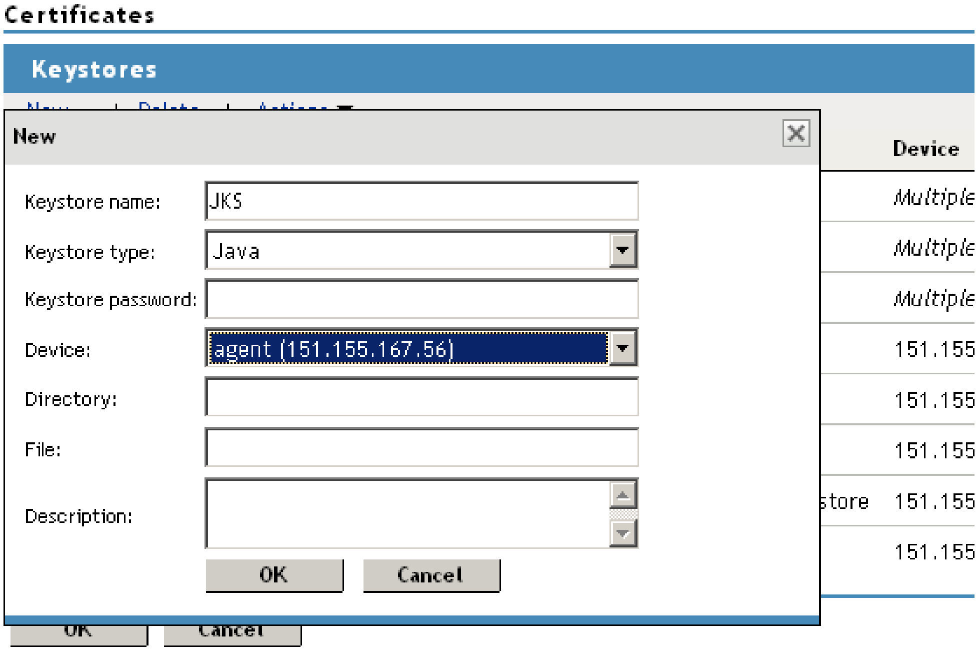
A keystore is storage file containing keys, certificates, and trusted roots. Access Manager agents can access them to retrieve certificates, keys, and trusted roots as needed. A trust store is a keystore containing only trusted roots. Intermediate CAs and end entity public certificates can be part of a trust store.

Access Manager comes with predefined stores for certificate management. However, in certain situations you might need to create a keystore or trust store. For example, if you are using JBoss keystore certificates that you need to import into Access Manager, you must create a keystore and assign it to the JBoss agent. It is probable that the keystore already exists on the JBoss file system, as created and configured by JBoss. Creating it again through Access Manager does not delete the existing keystore. This does allow Access Manager to recognize the existing keystore and add or remove the certificates. Access Manager cannot manage certificates that were created before the keystore is created in Access Manager.

The easiest way to create a keystore is to do so when you are adding the certificate to the keystore. If you want to create a trust store, the steps are identical, except you select trusted roots from the Trusted Roots page, rather than the certificates from the Certificates page.

A keystore stores only one certificate at a time. When you replace a certificate, you overwrite the existing one.

1. In the Administration Console, click *Certificates*.
2. Import the certificate, if you have not done so already. See Section 24.3, “Importing a Private/ Public Key Pair,” on page 367.
3. Click the certificate name.
4. In the Certificate Details page, click *Add Certificate to Keystores...*
5. On the Add Certificate to Keystores dialog box, click the *Select Keystore* button to browse for key stores.
6. On the Keystore page, click *New*.



Keystore name: JKS

Keystore type: Java

Keystore password:

Device: agent (151.155.167.56)

Directory:

File:

Description:

7. Fill in the following fields:

   Keystore name: Specifies the name of the keystore. This maps to a name that the server communication recognizes to identify the keystore on the device.

Keystore type: Specifies whether to use Java, PEM, or PKCS12.

Keystore password: Specifies the password to revise the keystore settings.

Device: Specifies the device (by IP) to which you assign the keystore. The device can be an Identity Server or SSL VPN. You cannot assign one keystore to multiple devices.

Directory: Specifies the directory where PKCS12 or PEM files are stored.

For example, `/var/opt/novell/keystores/`.

File: Specifies the path and filename of the Java keystore (JKS).

For example, `/var/opt/novell/keystores/myKeystore.keystore`.

Description: Describes the keystore.

8. Click *OK*.

   This creates the keystore.

9. (Optional) On the Keystore page, you can assign a certificate to the new keystore by selecting the store's check box.

10. Click *OK* in the *Add Certificate to Keystores* dialog box.

25.8 Reviewing the Command Status for Certificates

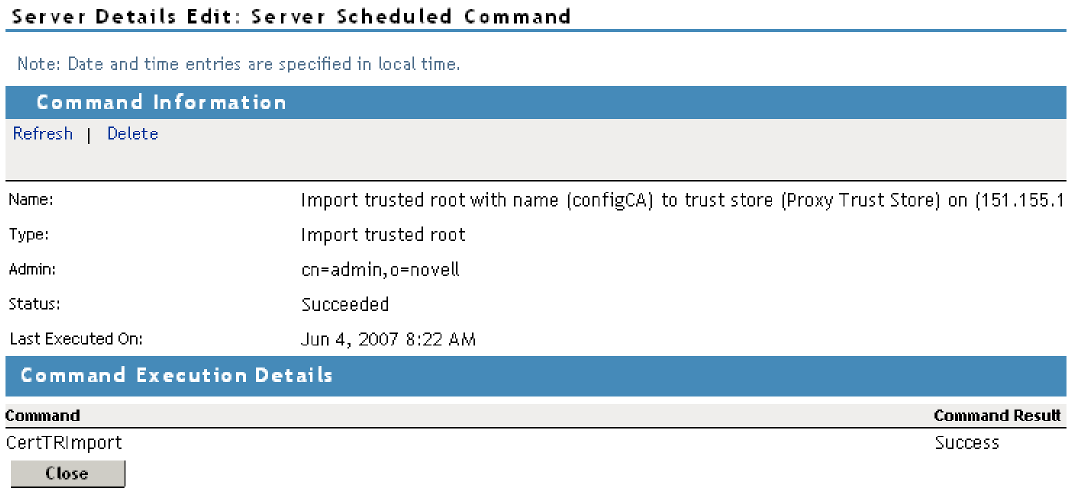
You can view the status of the commands that have been sent to the certificate server for execution. The following table describes this page:

1. In the Administration Console, click *Certificates*, then click *Command Status*.
2. Use the following options to review a server's certificate command status:
   - **Delete:** To delete a command, select the check box for the command, then click *Delete*. The selected command is cleared.
   - **Refresh:** Click *Refresh* to update the current cache of recently executed commands.
   - **Name:** Click this box to select all the commands in the list, then click *Refresh* or *Delete*.

   The following table describes the features on this page:

Column Name	Description
Name	Contains the display name of the command. Click the link to view additional details about the command.
Status	Specifies the status of the command. Some of the possible states of the command include `Pending`, `Incomplete`, `Executing`, and `Succeeded`.
Type	Specifies the type of server, such as Identity Server or Access Gateway.
Commands	Specifies the command given, such as `Import certificate`, or `Import trusted root`.
Admin	Specifies if the system or a user issued the command. If a user issued the command, the DN of the user is displayed.
Date & Time	Specifies the local date and time the command was issued.

3. To review command information click the link under a *Name*.



Server Details Edit: Server Scheduled Command

Note: Date and time entries are specified in local time.

Command Information

Name:	Import trusted root with name (configCA) to trust store (Proxy Trust Store) on (151.155.1
Type:	Import trusted root
Admin:	cn=admin,o=novell
Status:	Succeeded
Last Executed On:	Jun 4, 2007 8:22 AM

Command Execution Details

Command	Command Result
CertTRImport	Success

This page displays status information about the command and allows you to perform the following tasks:

Refresh: Select this option to update the current cache of recently executed commands.

Delete: Select this option to clear the current cache of recently executed commands.

The following command information is listed:

Name: Specifies the display name that has been given to the command.

Type: Specifies the type of command.

Admin: Specifies whether the system or a user issued the command. If a user issued the command, the field contains the DN of the user.

Status: Specifies the status of the command, and includes such states as *Pending*, *Incomplete*, *Executing*, and *Succeeded*.

Last Executed On: Specifies when the command was issued. The date and time are displayed in local time. If the command failed, additional information is available.

For a command that the Administration Console can successfully process, the page displays a *Command Execution Details* section with the name of the command and the command results.

4 Click *Close*.

Policy Management

VI

This section describes how Access Manager uses policies to assign roles, to control access, and to enable single sign-on to resources that require credentials.

Managing Policies

26

Policies are logical and testable rules that you use to maintain order, security, and consistency within your Access Manager infrastructure. You can specify activation criteria, deactivation criteria, temporal constraints (such as time of day or subnet), identity constraints (such as user object attribute values), and additional separation-of-duty constraints. Identity information can come from any identity source (such LDAP, an Identity Vault, or a directory) or from the Access Manager's Identity Server, which provides full Liberty Alliance specification support and SAML 2.0 support. Identity is available throughout the determination of rights and permissions.

26.1 Selecting a Policy Type

Access Manager uses the policy type to define the context within which a policy is evaluated. Each type of policy differs in purpose, which in turn determines the conditions and actions that apply. For example, the conditions and actions of an Authorization policy are going to differ from the conditions and actions of an Identity Injection policy.

When you click *New* on the Policies page, the system displays the predefined policy types in a drop-down list. Each policy type represents the set of conditions and actions that are available. You then configure rules to determine user roles, make decision requests, and enforce authorization decisions. You can also set up policies with no conditions, allowing actions to always take place. As policies and conditions become complex, it can be simpler and more manageable to design policies with conditions that deny or restrict access to large groups of users, rather than setting up policies that permit access to certain users.

Access Manager has the following policy types:

- **Access Gateway: Authorization:** This policy type is used to permit or deny access to protected resources, such as Web servers. After you have set up the protected resource, you use the policy rules to define how you want to restrict access. For example, if a user is denied access to a resource, you can use the policy to redirect them to a URL where they can request access to the resource.
- **Access Gateway: Identity Injection:** This policy type evaluates the rules for Identity Injection, which retrieves identity data from a data source (user store) and forwards it to Web applications. Such a policy can enable single sign-on. After the user has authenticated, the policy supplies the information required by the resource rather than allowing the resource to prompt the user for the information.
- **Access Gateway: Form Fill:** This policy type is used to create a policy that automatically fills in the information required in a form, after the user has accessed the form once. Such a policy can enable single sign-on to resources that require form data before allowing access.

- **Identity Server: Roles:** This policy type evaluates rules for establishing the roles of an authenticated user. Roles are generated based on policy statements each time a user authenticates. Roles are placed into an Authentication Profile, which can be used as input in policies for Authorization or Identity Injection.
- **J2EE Agent: EJB Authorization:** This policy type allows you to create policies that protect Enterprise JavaBeans*. You can protect the entire bean or specific interfaces or methods.
- **J2EE Agent:Web Authorization:** This policy type allows you to create policies that protect the Web applications on a J2EE server.

26.2 Policy Performance

Authorization and Identity Injection policies allow you to select conditions, one of which is Roles for Current Users. If you have thousands of users accessing your resources, you might want to design most of your policies to use roles. Roles are evaluated when a user logs in, and the roles assigned to the user are cached as long as the session is active. When the user accesses a resource protected by a policy that uses role conditions, the policy can be immediately evaluated because the user's role values are available. This is not true for all conditions; the values for some conditions must be retrieved from the user store. For example, if the policy uses a condition with an LDAP attribute, the user's value must be retrieved from the LDAP user store before the policy can be evaluated. On a system with medium traffic, this delay won't be noticed. On a system with high traffic, the delay might be noticeable.

However, you can design your policies to have the same results without causing the retrieval of the LDAP attribute value at resource access. You can create a Role policy for the LDAP attribute and have users assigned to this role at authentication when they match the attribute value requirements. Then when the users access the resources, they gain immediate access (or are immediately denied access) because their role assignments are cached.

If the same LDAP attribute policy is used to grant access to multiple resources, the chance that the user notices a delay is slight. The first time a policy is evaluated for a user, the data required for the policy is cached and therefore immediately available the next time it is requested. As you design your policies, experiment and find the type that works best for you and your customers.

26.3 Managing Policy Containers

You use policy containers to store and organize policies, similar to how you organize files in folders. The *Master_Container* is a permanent policy container, but you can use *Edit Policy Containers* to create new containers for purposes to suit your needs.

1. In the Administration Console, click *Access Manager > Policies*, then click the *Manage Policy Containers* icon by the *Policy Container* selection box.
2. On the Container List page, click *New*.
3. Name the policy container, then click *OK*.
4. Click *Close*.

After you add a policy container, the system displays it in the *Policy Container* drop-down list on the Policy List page.

If you have only one administrator configuring and managing policies, you can create additional policy containers to help you keep them organized. If you have multiple administrators creating policies, you can create a container for each administrator to use. This allows multiple administrators to modify policies at the same time. When an administrator opens a policy in a container, the container is locked, which prevents other administrators from modifying any policies in that container.

You must delete all the policies in a policy container before you can delete the policy container.

26.4 Managing Policies

- Section 26.4.1, “Creating Policies,” on page 387
- Section 26.4.2, “Deleting Policies,” on page 387
- Section 26.4.3, “Sorting Policies,” on page 387
- Section 26.4.4, “Importing and Exporting Policies,” on page 388

26.4.1 Creating Policies

Before creating policies, you need to design your policy strategy. For example, if you are going to use role-based access, you need to decide which roles you need and which roles allow access to your protected resources. Roles, which are used by Authorization policies that grant and deny access, need to be created first. If you have already created the roles and assigned them to users in your LDAP user store, you can use the values of your role attributes in the Authorization policies rather than using Access Manager roles.

To create a policy, see the following sections:

- Chapter 27, “Creating Role Policies,” on page 391
- Chapter 28, “Creating Authorization Policies,” on page 413
- Chapter 29, “Creating Identity Injection Policies,” on page 467
- Chapter 30, “Creating Form Fill Policies,” on page 481

26.4.2 Deleting Policies

A policy cannot be deleted as long as a resource is configured to use the policy. For Access Gateway and J2EE Agent policies, this means that you must remove the policy from all protected resources.

Roles can be used by Authorization, Form Fill, and Identity Injection policies. Before you can delete a Role policy, you must remove any reference to the role from all other policies.

26.4.3 Sorting Policies

Policies can be sorted by name and by type. On the Policies page, click *Name* in the *Policy List*, and the policies are sorted alphabetically by name. To sort alphabetically by type, click Type in the Policy List.

26.4.4 Importing and Exporting Policies

Policies which are created in the Administration Console can be exported and used in another Administration Console that is managing a different group of Access Gateways and other devices. Each policy type has slightly different import requirements. See the following:

- Section 27.6, "Importing and Exporting Role Policies," on page 412
- Section 28.8, "Importing and Exporting Authorization Policies," on page 465
- Section 29.8, "Importing and Exporting Identity Injection Policies," on page 478
- Section 30.5, "Importing and Exporting Form Fill Policies," on page 501

26.5 Managing a Rule List

You configure rules to create a policy. The rules collectively represent a desired course of action when the required conditions are met, such as denying entry-level employees access to a secure Web site, and permitting access for employees who have a role of Manager.

When the system evaluates the policy conditions, it begins with the rule with the highest priority and evaluates the conditions, starting with the first condition group in the rule. Each rule contains one or more conditions and one or more actions. If a rule's conditions are met, the rule's action is performed. For some policy types, the performance of any rule's action terminates the policy evaluation. With Authorization policies, for example, after the policy has determined that a user is either permitted or denied access to a resource, there is no reason to evaluate the policy further. However, a Role policy might identify multiple roles to which a user belongs. In this case, each rule of the policy must be evaluated to determine all roles to which the user belongs.

IMPORTANT: The interface for the policy engine is designed for flexibility. It does not protect you from creating rules that do nothing because they are always true or always false. For example, you can set up a condition where Client IP is equal to Client IP, which is always true. You are responsible for defining the condition so that it does a meaningful comparison.

You use rules to coordinate how a policy operates, and the behavior varies according to the policy type:

- Section 26.5.1, "Rule Evaluation for Role Policies," on page 388
- Section 26.5.2, "Rule Evaluation for Authorization Policies," on page 389
- Section 26.5.3, "Rule Evaluation for Identity Injection and Form Fill Policies," on page 389

26.5.1 Rule Evaluation for Role Policies

A Role policy is used to determine which role or roles a user is assigned to. However, you can specify only one role per rule. Role policies are evaluated when a user authenticates. Role policies do not directly deny or allow access to any resource, nor do they determine if a user is authenticated. A user's role can be used in the evaluation of an Authorization policy, but at that point the evaluation of the role policy has already occurred and is not directly part of the authorization process. The performance of an action (assigning a user to a role) does not terminate the evaluation of the policy, so subsequent rules in the Authorization policy continue to be evaluated.

26.5.2 Rule Evaluation for Authorization Policies

When the Access Gateway discovers a rule in an Authorization policy that either permits or denies a user access to a protected resource, it stops processing the rules in the policy. Use the following guidelines in determining whether your Authorization policy needs multiple rules:

- If the policy enforces multiple access requirements that can result in differing actions (either permit or deny), use separate rules to define the conditions and actions.
- If you want other conditions or actions processed when a rule fails, you must create a second rule for the users that fail to match the conditions.

If you create multiple rules, you can modify the order that the rules are processed. This allows you to create policies that contain a number of Permit rules that allow access if the user matches the rule. The lowest priority rule in such a policy is a Deny rule, which denies access to everyone who has not previously matched a Permit rule.

IMPORTANT: If you create policies with multiple Permit rules, you should make the last rule in the policy a generic deny policy (a rule with no conditions and with an action of deny). This ensures that if the Result on Error Condition field in a rule is set incorrectly, the user matches the last rule and is denied access. Without this rule, a user might gain access because the user didn't match any of the rules.

You can also create a number of policies and enable multiple policies for the same protected resource. Rule priority determines how the enabled policies interact with each other. The rules in the policies are gathered into one list, then sorted by priority. The processing rules are applied as if the rules came from one policy. It is a personal design issue whether you create a policy with multiple rules or create multiple policies that you enable on a single protected resource. Either design produces a list of rules, sorted by priority, that is applied to the user requesting access to the protected resource.

26.5.3 Rule Evaluation for Identity Injection and Form Fill Policies

Rules in Identity Injection and Form Fill policies have actions, but no conditions. Because they have no conditions, all the rules are evaluated and the actions are performed. Identity Injection policies have two exceptions to this rule; they can insert only one authentication header and one cookie header. If you create multiple rules, each with an authentication header and a cookie header, the rule with the highest priority is processed and its actions performed. The actions in the second rule for injecting an authentication header and a cookie header are ignored.

You cannot create multiple rules for a Form Fill policy.

26.6 Enabling Policy Logging

Policy logging is expensive; it uses processing time and disk space. In a production environment, you should enable it only under the following types of conditions:

- You have created a new policy and need to verify its functionality.
- You are troubleshooting a policy that is not behaving as expected.

To gather troubleshooting information, you should enable the *File Logging* and *Echo To Console* options in the Identity Server configuration and set the *Component File Logger Levels* for *Application* to at least *info*. Then you must update the Identity Server configuration and restart any Access Gateway ESPs, so that the ESPs read the logging options. See Section 32.2, “Configuring Identity Server Logging,” on page 516. When you have solved the problem, you should disable these options.

The log file on the component that executed the policy is where you should look for logging information. For example, if you have an Access Gateway: Authorization error, look at the log on the Access Gateway that executed the policy.

For additional policy troubleshooting procedures, see Chapter 39, “Troubleshooting Access Manager Policies,” on page 599.

Creating Role Policies

27

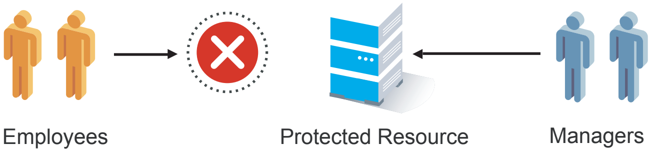
This section describes the following topics for Identity Server roles.

- Section 27.1, "Understanding RBAC in Access Manager," on page 391
- Section 27.2, "Creating Roles," on page 394
- Section 27.3, "Creating Access Manager Roles from an Existing Role-Based Policy System," on page 403
- Section 27.4, "Mapping Roles between Trusted Providers," on page 410
- Section 27.5, "Enabling and Disabling Role Policies," on page 411
- Section 27.6, "Importing and Exporting Role Policies," on page 412

27.1 Understanding RBAC in Access Manager

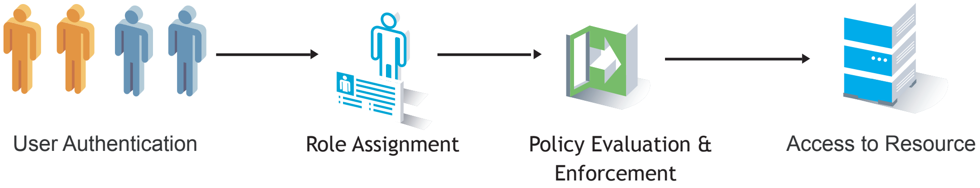
Role-based access control (RBAC) provides a convenient way to assign a user to a particular job function or set of permissions within an enterprise, in order to control access. As an administrator, you probably have defined a set of roles for your needs. Your roles might include Employee, Student, Administrator, Manager, and so on. You might have Web resources that you want available to all employees, or only to managers, as shown in Figure 27-1.

Figure 27-1 *Traditional RBAC*



Access Manager supports core RBAC functionality by providing user role mapping and the mapping of roles to resource rights and permissions. User role mapping is a primary function of a Role policy. Role mapping to resource rights is accomplished through Authorization policies and role settings in J2EE and SSL VPN environments. When creating a role, you assign users to the role, based on attributes of their identities. You also specify the constraints to place on the role.

Figure 27-2 *RBAC Using a Policy*



As shown in Figure 27-2, during user authentication, the system checks the existing Role policy to determine which roles that a user must be assigned to. After authentication, assigned roles can be used as evaluated conditions of an Authorization policy.

Java applications and Web server applications can also be configured to use roles for access control. For these applications you can use Access Manager to assign the users to the required roles. You can then use the J2EE agent to forward the user's assigned roles to the Java application, or use Access Gateway Identity Injection policies to inject the assigned roles into the HTTP header that is sent to the Web server.

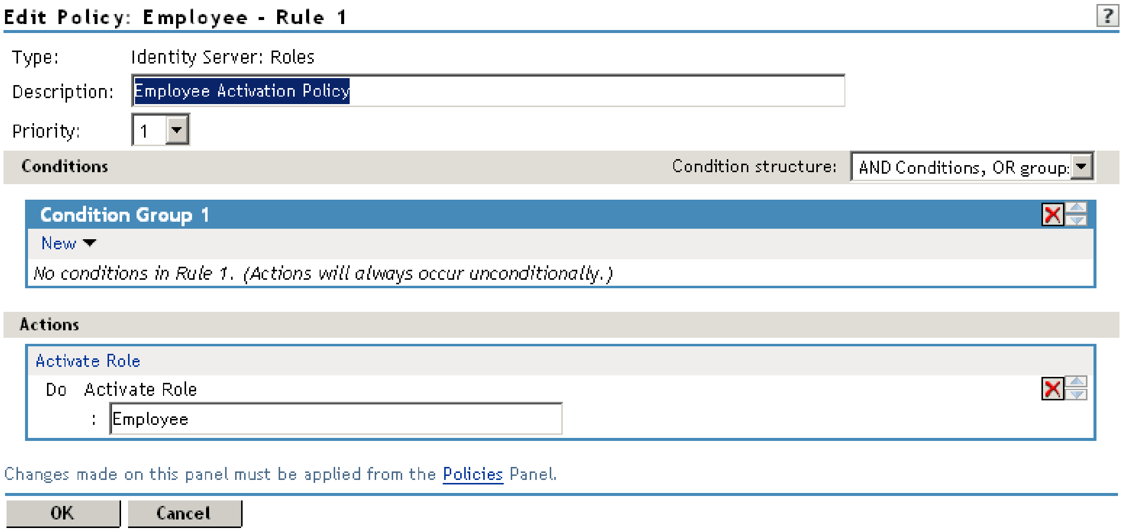
The following examples describe ways to use roles in Access Manager.

- Section 27.1.1, "Assigning all Authenticated Users to a Role," on page 392
- Section 27.1.2, "Using a Role to Create an Authentication Policy," on page 392
- Section 27.1.3, "Using Prioritized Rules in an Authorization Policy," on page 394

27.1.1 Assigning all Authenticated Users to a Role

The system assigns users to roles when they authenticate. The following example illustrates a Role policy that creates an Employee role. All authenticated users are assigned to the role of Employee, because it does not include any conditions (see "Employee Role" on page 399).

Figure 27-3 *Employee Role Policy*



Edit Policy: Employee - Rule 1

Type: Identity Server: Roles

Description: Employee Activation Policy

Priority: 1

Conditions

Condition structure: AND Conditions, OR group

Condition Group 1

No conditions in Rule 1. (Actions will always occur unconditionally.)

Actions

Activate Role

Do Activate Role : Employee

Changes made on this panel must be applied from the Policies Panel.

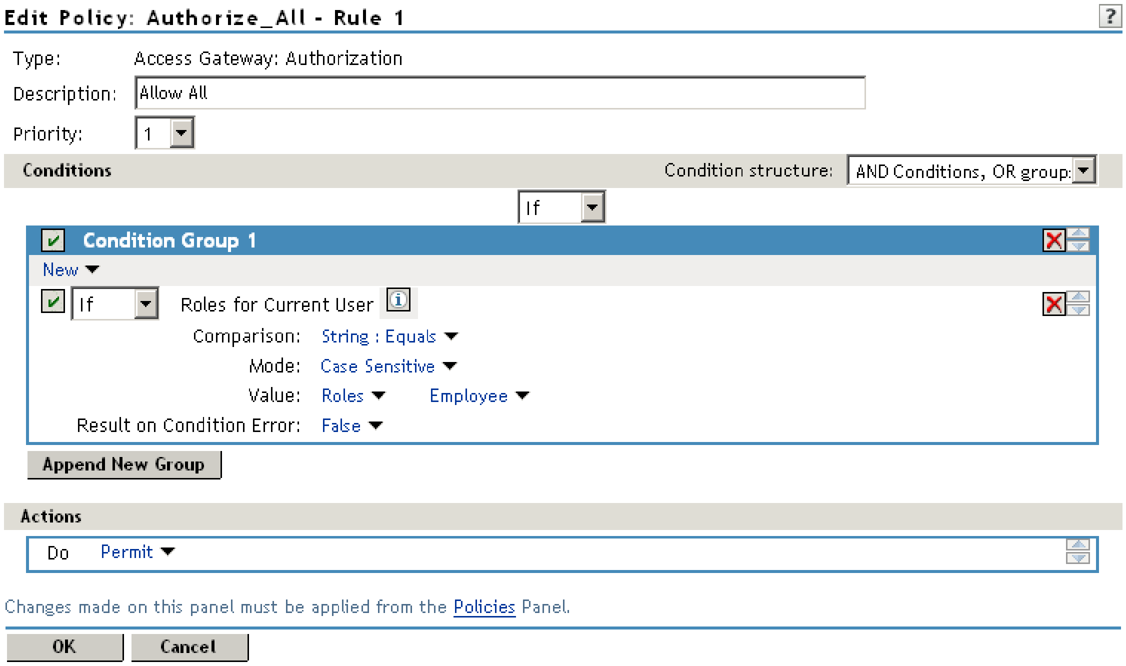
Role assignment audit events can be created during authentication to the Identity Server. You enabled this on the Logging page in the Identity Server configuration when you enable the *Login Provided* or *Login Consumed* options.

27.1.2 Using a Role to Create an Authentication Policy

The simplest implementation of RBAC policies is to include roles as evaluated conditions when creating Authorization policies.

Suppose you belong to a company of 300 employees, and ten of them are managers. You can assign all employees to an Employee role, and make it a condition of an Authorization policy with no restrictions. Such a policy would permit access to Web resources intended for all employees, as shown in the following example:

Figure 27-4 *Employee Authorization Policy*



Edit Policy: Authorize_All - Rule 1

Type: Access Gateway: Authorization

Description: Allow All

Priority: 1

Conditions

Condition structure: AND Conditions, OR group:

If

Condition Group 1

If Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles Employee

Result on Condition Error: False

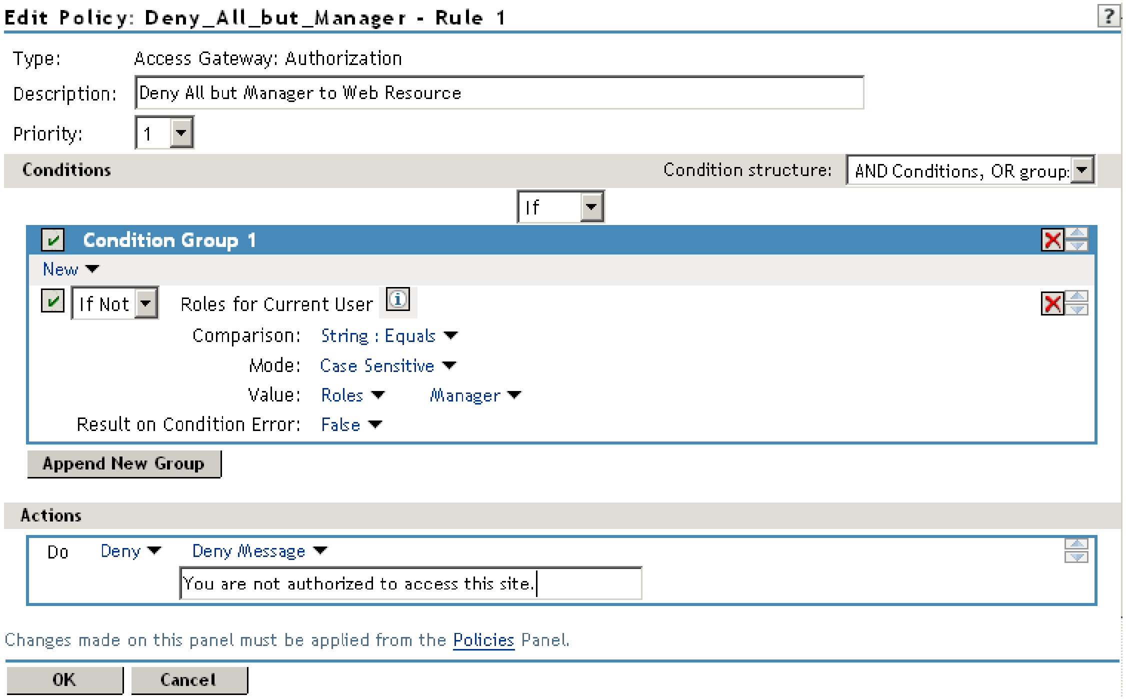
Actions

Do Permit

Changes made on this panel must be applied from the Policies Panel.

For more sensitive Web resources intended only for managers, you might create a role called Manager. (See "Manager Role" on page 401). The Manager role might be a condition of an Authorization policy that denies access to any employee that has not been assigned to the Manager role when the user authenticated. The following example illustrates this. Notice that the operand for the governing condition logic is set to `If Not`.

Figure 27-5 *Manager Authorization Policy*



Edit Policy: Deny_All_but_Manager - Rule 1

Type: Access Gateway: Authorization

Description: Deny All but Manager to Web Resource

Priority: 1

Conditions

Condition structure: AND Conditions, OR group:

If

Condition Group 1

If Not Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles Manager

Result on Condition Error: False

Actions

Do Deny Deny Message

You are not authorized to access this site.

Changes made on this panel must be applied from the Policies Panel.

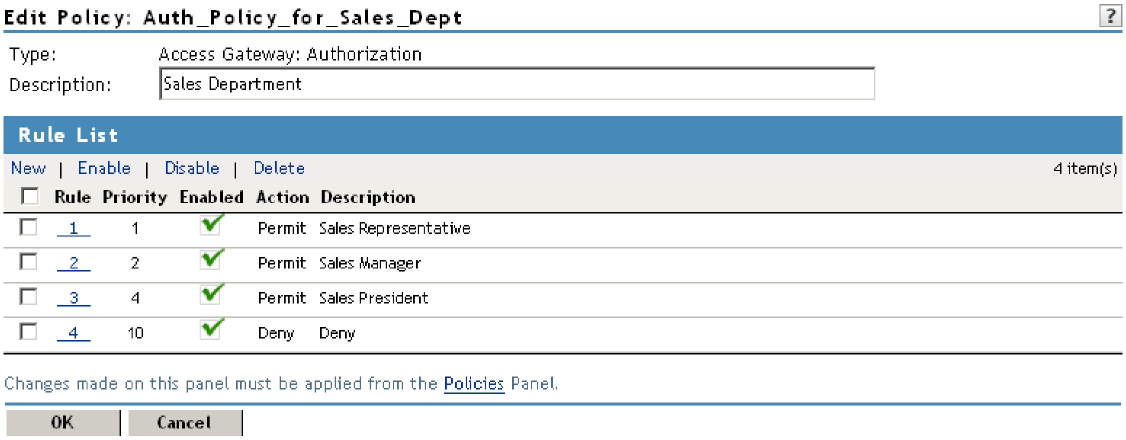
After you have created the Authorization policies, you need to assign the policies to the resources they were designed to protect.

See Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210, “Assigning a Web Authorization Policy to the Resource”, and “Assigning an Enterprise JavaBeans Authorization Policy to a Resource”.

27.1.3 Using Prioritized Rules in an Authorization Policy

In another policy example, you might create an Authorization policy for the Sales Department and set up a list of rules that evaluate whether a user has been assigned to one of the roles associated with the department, and then deny access if the user has not been assigned to any of them, as shown in the Rule List page for the Authorization policy below:

Figure 27-6 *Authorization Policy with Multiple Rules*



Edit Policy: Auth_Policy_for_Sales_Dept

Type: Access Gateway: Authorization

Description: Sales Department

Rule List

☐	Rule	Priority	Enabled	Action	Description
☐	1	1	✓	Permit	Sales Representative
☐	2	2	✓	Permit	Sales Manager
☐	3	4	✓	Permit	Sales President
☐	4	10	✓	Deny	Deny

Changes made on this panel must be applied from the Policies Panel.

In this example, you specify a first-priority rule with a condition that allows access if a user has been assigned to the role of Sales Representative. You add rules for users assigned to the a role of Sales Manager, Sales Vice President, and so on. You then create a lowest-priority rule that contains no conditions, and an action of Deny. This policy denies any user who has not been assigned a Sales department role. When users do not meet the conditions of the rules, the user is denied access by the lowest-priority rule.

For more information on using roles in Authorization policies, see Chapter 28, “Creating Authorization Policies,” on page 413.

27.2 Creating Roles

To implement RBAC, you must first define all of the roles within your organization and the permissions attached to each role. A collection of users requiring the same access can be assigned to a single role. Each user can also be assigned to one or more roles and receive the collective rights associated with the assigned roles. A role policy consists of one or more rules, and each rule consists of one or more conditions and an action.

The following topics discuss how to create a role.

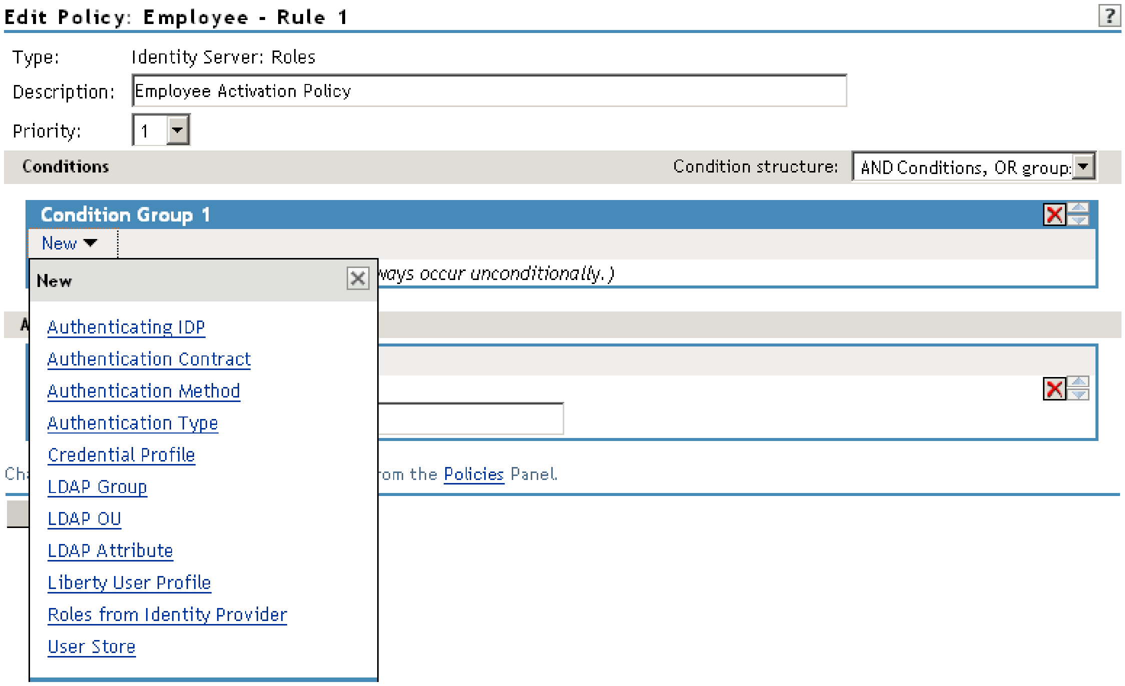
- Section 27.2.1, “Selecting Conditions,” on page 395
- Section 27.2.2, “Selecting an Action,” on page 398

- Section 27.2.3, “Reviewing the Rules,” on page 398
- Section 27.2.4, “Example Role Policies,” on page 399

27.2.1 Selecting Conditions

You create a role by selecting the appropriate conditions that qualify a user to be assigned to a role, as shown in the following page.

Figure 27-7 *Role Policy Conditions*



Edit Policy: Employee - Rule 1

Type: Identity Server: Roles

Description: Employee Activation Policy

Priority: 1

Condition structure: AND Conditions, OR group:

Table 27-1 describes the conditions available for a Role policy:

Table 27-1 *Possible Role Conditions*

Role Condition	Description
Authenticating IDP	Specifies the identity provider that authenticated the current user. To use this condition, you must have set up a trusted relationship with more than one identity provider. See Chapter 9, “Configuring Trusted Providers,” on page 139. The most common way to use this condition is when you have a service provider that has been configured to trust two identity providers and you want to assign a role based on which identity provider authenticated the user. To configure such a policy: ◆ Set the *Authenticating IDP* field to *[Current]* ◆ Set the *Value* field to *Authenticating IDP* ◆ Select the name of an identity provider For the condition to evaluate to True, the identity provider specified in the policy must be the one that the user selected for authentication.
Authentication Contract	Specifies the contract used to authenticate the current user. The selections in this list are defined in the Identity Server configuration (*Local* > *Contracts*). The *Comparison* value can be an exact string, the start, the end, or a substring. The most common way to use this condition is to select *[Current]* for the *Authentication Contract* field and to select *Authentication Contract* and the name of a contract for the *Value* field. If you select *Data Entry Field* for the *Value* field, you must specify the URI of the contract for the conditions to match. For a list of these values, click *Access Manager* > *Identity Servers* > *Edit* > *Local* > *Contracts*.
Authentication Method	Specifies the method used to authenticate the current user.
Authentication Type	Compares a selected authentication type to the authentication types used to authenticate the current user. *[Current]* represents the current set of authentication types used to authenticate the user. The other selections represent specific authentication types that can be used to compare with *[Current]*. The *Authentication Type* condition returns True if the selected authentication type is contained in the set of authentication types for *[Current]*. For example, if the current user was required to satisfy the authentication types of Basic and SmartCard, then a selected authentication type of either Basic or SmartCard would match.

Role Condition	Description
Credential Profile	Specifies the credentials used by the user during authentication. Only values used at authentication time are available for this comparison. The *Comparison* value can be an exact string, the start, the end, or a substring. The default contracts assign the cn attribute to the Credential Profile. If you create your own authentication contract, you can assign a different attribute to the Credential Profile. If your user store is an Active Directory server, you need to be aware that the cn attribute is used even though the user login is chosen from the SAMAccountName attribute. If you want to use the SAMAccountName attribute in the Credential Profile, you need to create your own authentication contract. If you select *Data Entry Field* as the *Value* type, be aware of the following requirements: ◆ If you selected *LDAP User DN* as the credential, you need to specify the DN of the user in the *Value* text box. If the comparison type is set to *Contains Substring*, you can match a group of users by specifying a common object that is part of their DNs, for example ou=sales. ◆ If you selected *X509 Public Certificate Subject* as the credential, you need to specify all elements of the Subject Name of the certificate in the *Value* text box. Separate the elements with a comma and a space, for example, o=novell, ou=sales. If the comparison type is set to *Contains Substring*, you can match a group of certificates by specifying a name that is part of their Subject Name, for example ou=sales.

LDAP Group	Specifies a group in which the authenticating user is evaluated for membership. The value, an LDAP DN, must be a fully distinguished name of a group.
	If the Administration Console and the Identity Server are installed on separate machines, you need to specify *Data Entry Field* for the *Value* and enter the fully distinguished name of the group in the text box. For example:
	`cn=managers,cn=users,dc=bcf2,dc=provo,dc=novell,dc=com`
	`or`
	`cn=manager,o=novell`

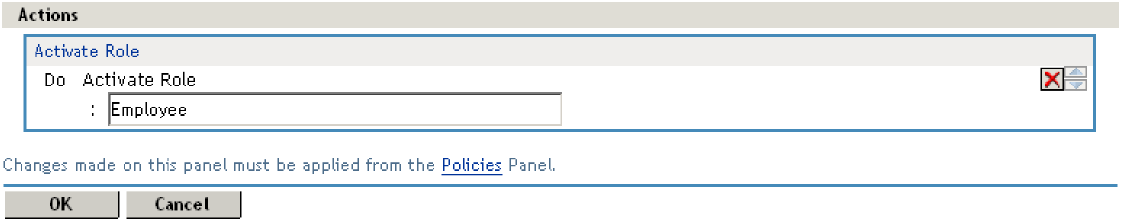
LDAP OU	Specifies an OU for which the authenticating user's DN (distinguished name) is evaluated for containment. The value, an LDAP DN, must be a fully distinguished name of an organizational unit.
	If the Administration Console and the Identity Server are installed on separate machines, you need to specify *Data Entry Field* for the *Value* and enter the fully distinguished name of the organizational unit in the text box. For example:
	`cn=users,dc=bcf2,dc=provo,dc=novell,dc=com`
	`or`
	`ou=users,o=novell`

Role Condition	Description
LDAP Attribute	Specifies an attribute from the user object of an authenticated user. By default the selection values include those defined by inetOrgPerson.
Liberty User Profile	Specifies any one of a number of data values that have been mapped to a Liberty Profile attribute. To check the mapping of attribute values, click *Identity Servers* > *Servers Edit* > *Liberty* > *LDAP Attribute Mapping*.
Roles from Identity Provider	Roles that are passed from an identity provider to another trusted provider. For example, a service provider requests authentication from an identity provider, which returns a role. This role might be passed to the protected resource as a condition for authorization. This condition uses the mapped attribute All Roles. All roles that are assigned to the user can be mapped to Liberty and SAML 2 attributes and assigned to a trusted identity provider. (See Section 9.8, "Selecting Attributes for a Trusted Provider," on page 155 for information about enabling All Roles.) For an example of how to use *Roles from Identity Provider*, see Section 27.4, "Mapping Roles between Trusted Providers," on page 410.
User Store	Compares a selected user store with the user store from which the current user is authenticated. The *[Current]* selection represents the user store from which the user was authenticated. The other selections represent all of the configured user stores that can be used to compare with *[Current]*. For example, if the configured user stores are eDir1 and AD1 and the current user is authenticated from eDir1, then a selected user store of eDir1 would match and a selected user store of AD1 would not match.

27.2.2 Selecting an Action

The policy action specifies the role to which the user belongs. Roles are activated at the time the role policy is evaluated. In the following page, the role of Employee is specified to be assigned to the user.

Figure 27-8 *Assigning a Role*



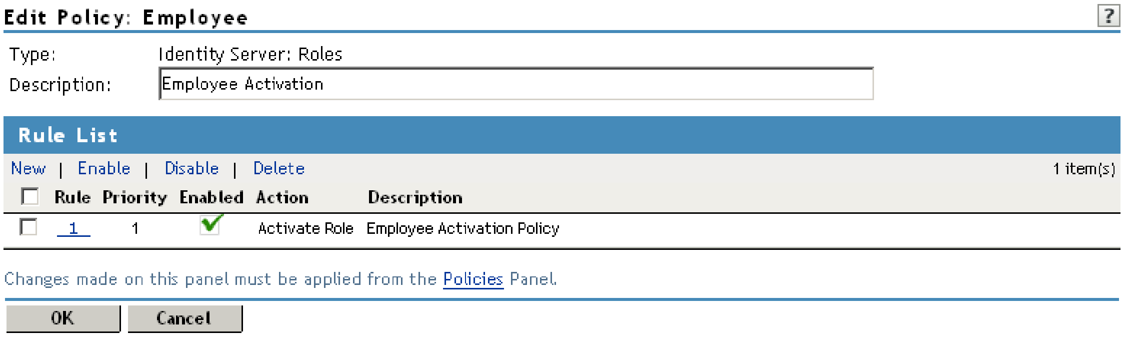
Do Activate Role
: Employee

Changes made on this panel must be applied from the Policies Panel.

27.2.3 Reviewing the Rules

After you create roles, they are displayed as rules on the Edit Policy page, where you can review the priority, action, and a description of the role, as shown in the following page.

Figure 27-9 *Rule Summary*



Edit Policy: Employee

Type: Identity Server: Roles

Description: Employee Activation

Rule List

☐	Rule	Priority	Enabled	Action	Description
☐	1	1	✓	Activate Role	Employee Activation Policy

Changes made on this panel must be applied from the Policies Panel.

27.2.4 Example Role Policies

The following instructions describe how to create two types of roles: a general Employee role and a restrictive Manager role. These roles can be used by the Access Gateway in Identity Injection policies and by the Access Gateway and the J2EE Agent in Authorization policies.

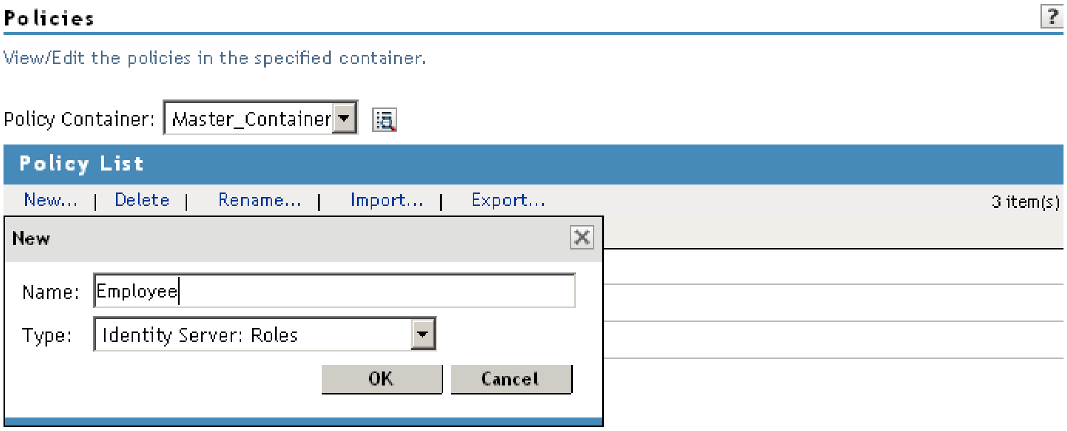
- "Employee Role" on page 399
- "Manager Role" on page 401

Employee Role

This role policy creates an Employee role. All authenticated users are assigned to this role when they log in (because it does not include conditions). This role can then be used to grant access to resources to all users in your user stores.

1 In the Administration Console, click *Access Manager > Identity Servers > Servers > Edit > Roles > Manage Policies.*

2 On the Policies page, click *New*.



Policies

View/Edit the policies in the specified container.

Policy Container: Master_Container

Name: Employee

Type: Identity Server: Roles

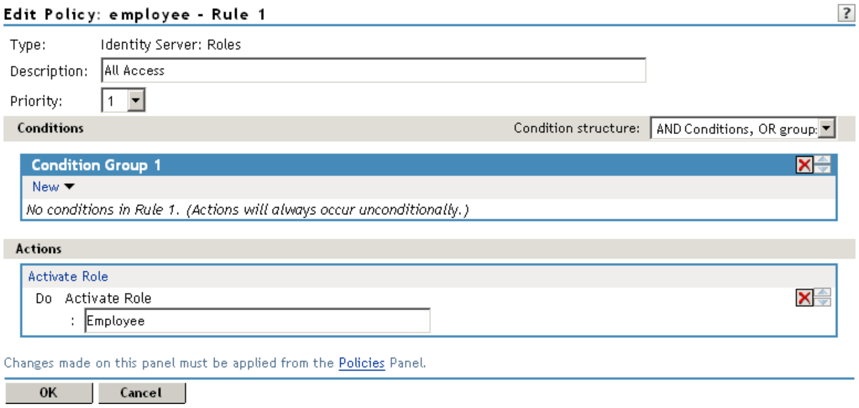
3 Select a policy type of *Identity Server: Roles* and specify a display name, such as Employee.

4 Click *OK*.

5 On the Edit Policy page, specify a description in the *Description* field.

It is important to use this field to keep track of your roles and policies. The policy feature is powerful, and setup can be as large and complex as you want it to be, with a potentially unlimited number of conditions and choices. This description is useful to help keep track of various role and policy configurations.

6 Make sure the *Condition Group 1* section has no conditions, so that all users who authenticate match the condition.



Edit Policy: employee - Rule 1

Type:	Identity Server: Roles
Description:	All Access
Priority:	1

Conditions — Condition structure: AND Conditions, OR groups

Condition Group 1

No conditions in Rule 1. (Actions will always occur unconditionally.)

Actions

Activate Role

Do Activate Role : Employee

Changes made on this panel must be applied from the Policies Panel.

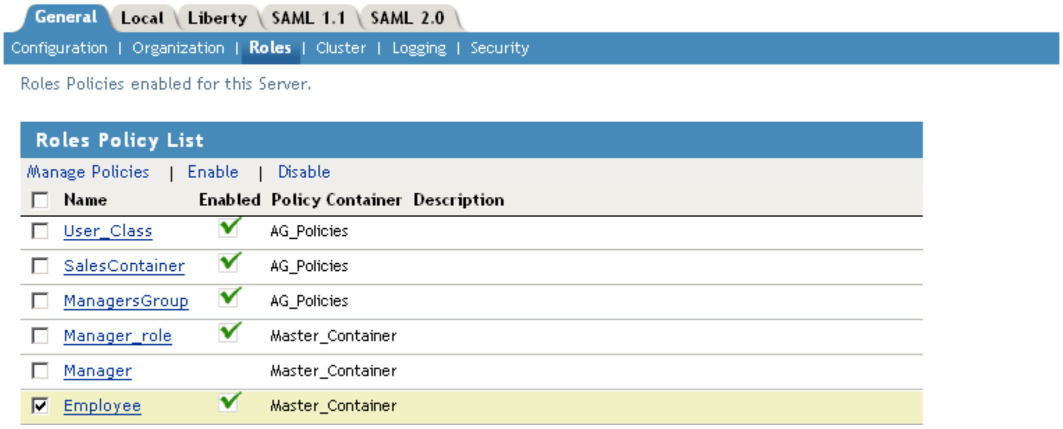
7 In the *Actions* section, click *Activate Role*.

8 In the *Activate Role* box, type `Employee`, then click *OK*.

If this role needs to match the name of a role required by a Java or Web application, ensure that the case of the name matches the application's name.

9 On the Edit Policy (Rule List) page, click *OK*.

10 On the Policies page, click *Apply Changes*, then click *Close*.



General
Local
Liberty
SAML 1.1
SAML 2.0
Configuration | Organization | Roles | Cluster | Logging | Security

Roles Policies enabled for this Server.

Roles Policy List

Manage Policies | Enable | Disable

☐	Name	Enabled	Policy Container	Description
☐	User_Class	✓	AG_Policies	
☐	SalesContainer	✓	AG_Policies	
☐	ManagersGroup	✓	AG_Policies	
☐	Manager_role	✓	Master_Container	
☐	Manager		Master_Container	
☑	Employee	✓	Master_Container	

11 On the Role Policy page, select the Employee role, then click *Enable.*

12 On the *Servers* tab, click *Update Servers*.

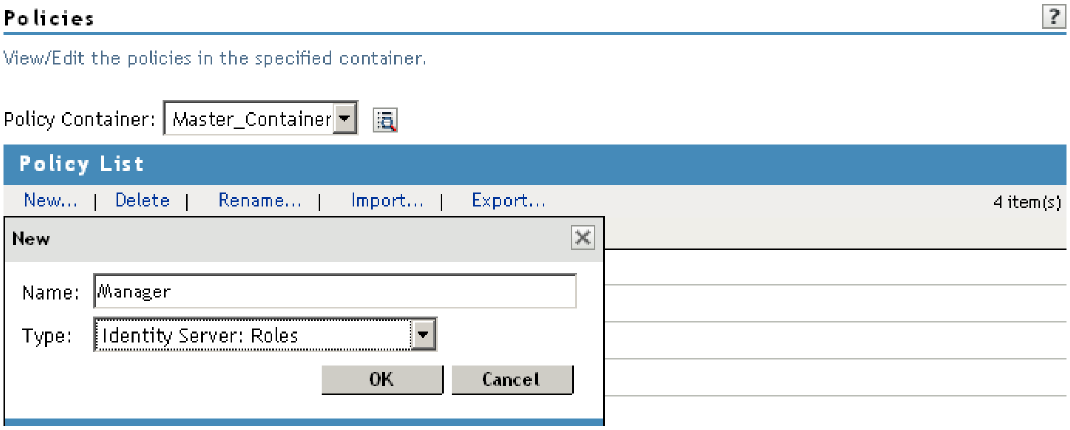
This step updates the Identity Server configuration, which is required after you create a role.

13 To create a Manager role, continue with "Manager Role" on page 401.

Manager Role

Because the Manager role is restrictive, role policy conditions must be specified. The Manager role is assigned only to the users who meet the conditions.

1. Click *Identity Servers > Servers > Edit > Roles > Manage Policies.*
2. On the Policies page, click *New*.



Policies

View/Edit the policies in the specified container.

Policy Container: Master_Container

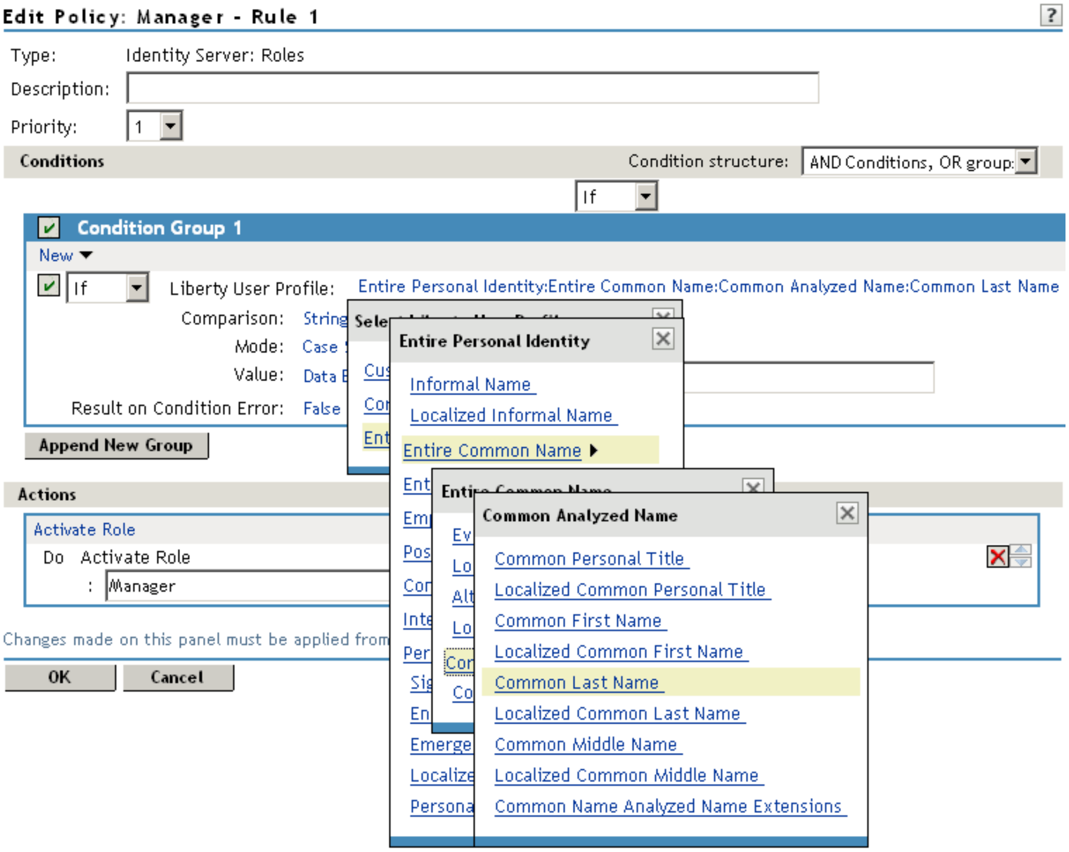
Policy List

New

Name: Manager

Type: Identity Server: Roles

3. Select a policy type of *Identity Server: Roles* and specify a display name (for this example, Manager.)
4. Click *OK*.
5. In the *Conditions* section, click *New > Liberty User Profile*.



Edit Policy: Manager - Rule 1

Type: Identity Server: Roles

Description:

Priority: 1

Conditions

Condition structure: AND Conditions, OR group:

If

Condition Group 1

If Liberty User Profile: Entire Personal Identity:Entire Common Name:Common Analyzed Name:Common Last Name

Comparison: String

Mode: Case

Value: Data E

Result on Condition Error: False

Actions

Activate Role

Do Activate Role

: Manager

Changes made on this panel must be applied from

6 In the *Condition Group 1*, select the conditions the user must meet:

Liberty User Profile: Select *Entire Personal Identity > Entire Common Name > Common Analyzed Name > Common Last Name.*

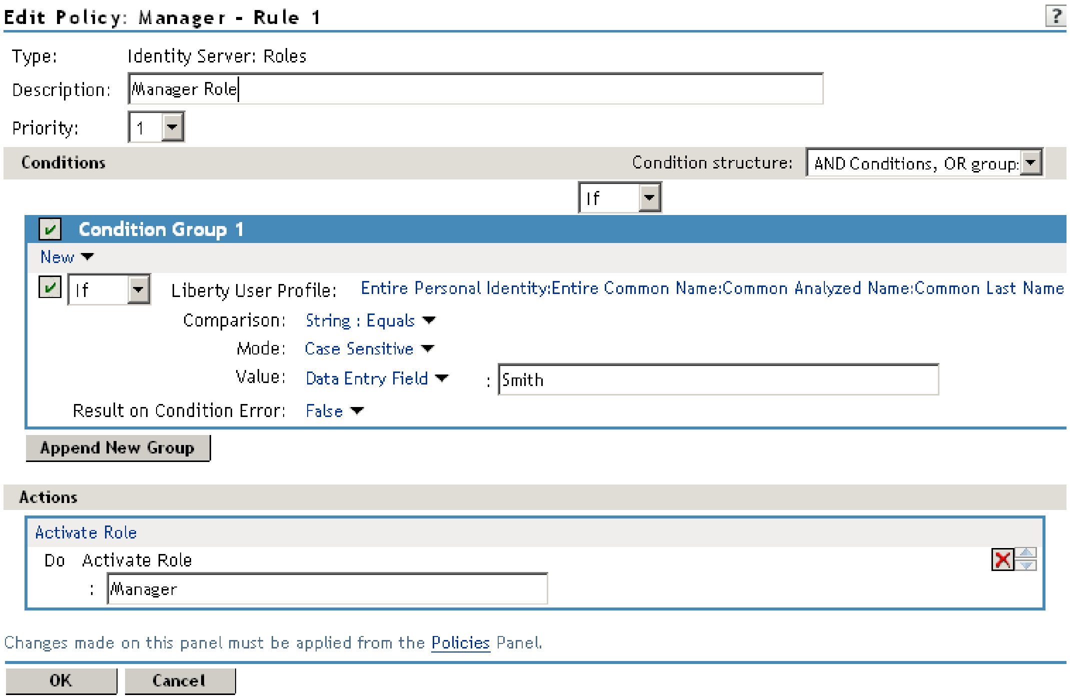
Comparison: Select how you want the attribute values to be compared. For the Common Last Name attribute, select *String > Equals*.

Mode: Select *Case Insensitive*.

Value: Select *Data Entry Field* and type the person's name in the box (Smith, in this example). This sets up the condition that if the user has the name Smith, his or her role as Manager is activated at authentication.

Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of Manager if the condition evaluates to *True*. If an error occurs, you do not want random users assigned the role of Manager. Therefore, for this rule, you need to select *False*.

7 In the *Actions* section, click *Activate Role*.



Edit Policy: Manager - Rule 1

Type: Identity Server: Roles

Description: Manager Role

Priority: 1

Conditions Condition structure: AND Conditions, OR groups

If

Condition Group 1

☑ If Liberty User Profile: Entire Personal Identity:Entire Common Name:Common Analyzed Name:Common Last Name

Comparison: String : Equals

Mode: Case Sensitive

Value: Data Entry Field : Smith

Result on Condition Error: False

Actions

Activate Role

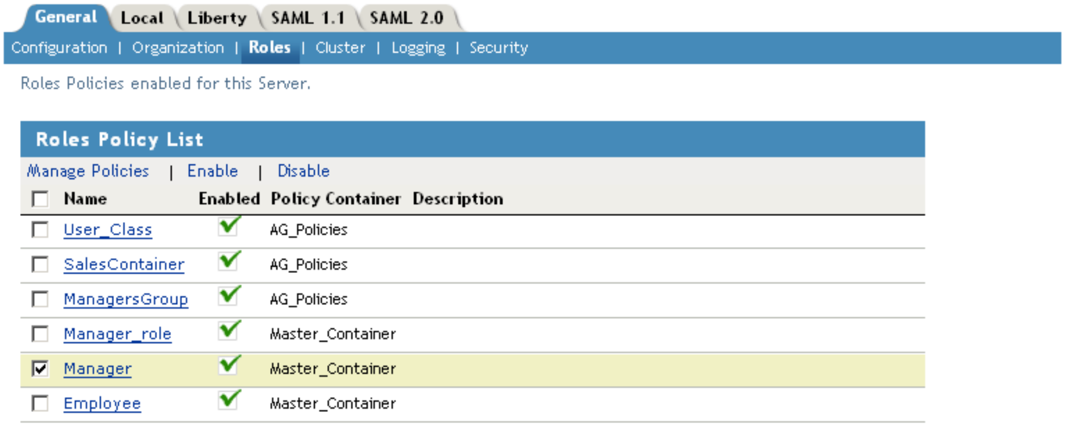
Do Activate Role

: Manager

Changes made on this panel must be applied from the Policies Panel.

8 In the *Activate Role* box, type `Manager`, then click *OK* twice.

9 On the Policies page, click *Apply Changes*.



General
Local
Liberty
SAML 1.1
SAML 2.0
Configuration | Organization | Roles | Cluster | Logging | Security

Roles Policies enabled for this Server.

Roles Policy List

Manage Policies | Enable | Disable

☐	Name	Enabled	Policy Container	Description
☐	User_Class	✓	AG_Policies	
☐	SalesContainer	✓	AG_Policies	
☐	ManagersGroup	✓	AG_Policies	
☐	Manager_role	✓	Master_Container	
☑	Manager	✓	Master_Container	
☐	Employee	✓	Master_Container	

10 Select the Manager role, then click *Enable*

11 On the *Servers* tab, click *Update Servers*.

27.3 Creating Access Manager Roles from an Existing Role-Based Policy System

If you have already implement a role-based administration policy for granting access to print, file, and LDAP resources, you can leverage your role definitions and use Access Manager policies to control access to Web resources. If your role definitions use the following types of LDAP features, you can create Access Manager Role policies that use them:

- The values found in LDAP attributes
- The location of the user objects in the directory tree
- Membership in groups or roles

The Access Manager Role policies that you create using these features can then be used to control access to protected Web resources.

- Section 27.3.1, “Creating a Role Using an LDAP Attribute,” on page 404
- Section 27.3.2, “Creating a Role Using the Location of the User Objects,” on page 405
- Section 27.3.3, “Creating a Role Using Role or Group Membership,” on page 408

27.3.1 Creating a Role Using an LDAP Attribute

You can assign a user to a role by using a value found in any LDAP attribute in your directory. The following example uses the objectClass attribute because every object in an LDAP directory has an objectClass attribute that contains the object classes to which the object belongs. This attribute contains the name of the object class that was used to create the object as well as the names of the superior object classes of this class. All you need to know is the name of the object class you used to create your users in the LDAP directory. For example, the following instructions create a Role policy for users who were created with the User object class.

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Click *New*, specify a name for the Role policy, select *Identity Server: Roles* for the type, then click *OK*.
3. In *Condition Group 1*, click *New*, then select *LDAP Attribute*.
4. In *Condition Group 1*, select the conditions the user must meet:

   LDAP Attribute: Select the objectClass attribute. If you have not added this attribute, it won’t appear in the list. Scroll to the bottom of the list, click *New LDAP Attribute*, specify objectClass for the name, then click *OK*.

   If you are using eDirectory™ for your LDAP directory, you need to specify standard LDAP names for the attributes. Access Manager does not support spaces or colons in attribute names.

   Comparison: Select how you want the attribute values to be compared. For the objectClass attribute, select *String* > *Contains Substring*.

   The objectClass attribute is a multi-valued attribute and, for most objects, contains multiple values. For example in eDirectory, users created with the User object class have User, organizationalPerson, person, ndsLoginProperties, and top as values in the objectClass attribute.

   Mode: Select *Case Insensitive*.

   Value: Select *Data Entry Field* and specify User as the value.

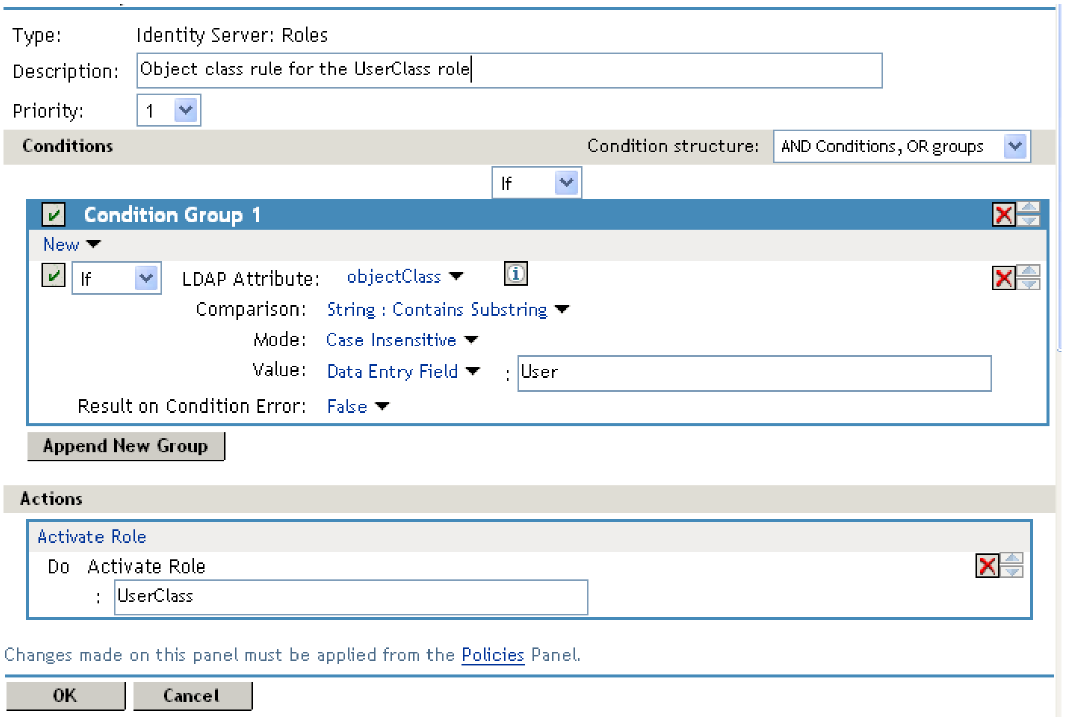
   Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of UserClass if the condition evaluates to *True*. If an error occurs, you do not want random users assigned the role of UserClass. Therefore, for this rule, you need to select *False*.

5 In the *Actions* section, click *Activate Role*.

6 In the *Activate Role* box, type `UserClass`, then click *OK*

The name you enter in the box is the role you want assigned to the users who match the condition.

Your rule should look similar to the following:



Type: Identity Server: Roles

Description: Object class rule for the UserClass role

Priority: 1

Conditions

Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

New

☑ If LDAP Attribute: objectClass

Comparison: String : Contains Substring

Mode: Case Insensitive

Value: Data Entry Field : User

Result on Condition Error: False

Append New Group

Actions

Activate Role

Do Activate Role

: UserClass

Changes made on this panel must be applied from the Policies Panel.

OK Cancel

7 Click *OK* twice, then click *Apply Changes*.

8 To enable the role so that it can be used in Authorization and Identity Injection policies, click *Identity Servers* > *Edit* > *Roles*.

9 Select the check box by the name of the role, then click *Enable*.

10 Click *OK*.

11 To update the Identity Server, click *Servers* > *Update Servers*.

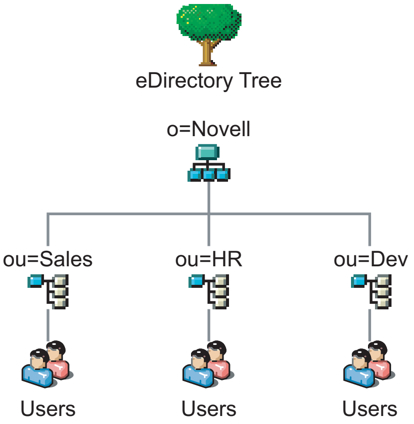
You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see the following:

- Chapter 28, "Creating Authorization Policies," on page 413
- Chapter 29, "Creating Identity Injection Policies," on page 467

27.3.2 Creating a Role Using the Location of the User Objects

If you have created your users in specific containers in your LDAP tree, you can use these container objects to assign users to roles. For example, suppose your LDAP tree looks similar to the following tree.

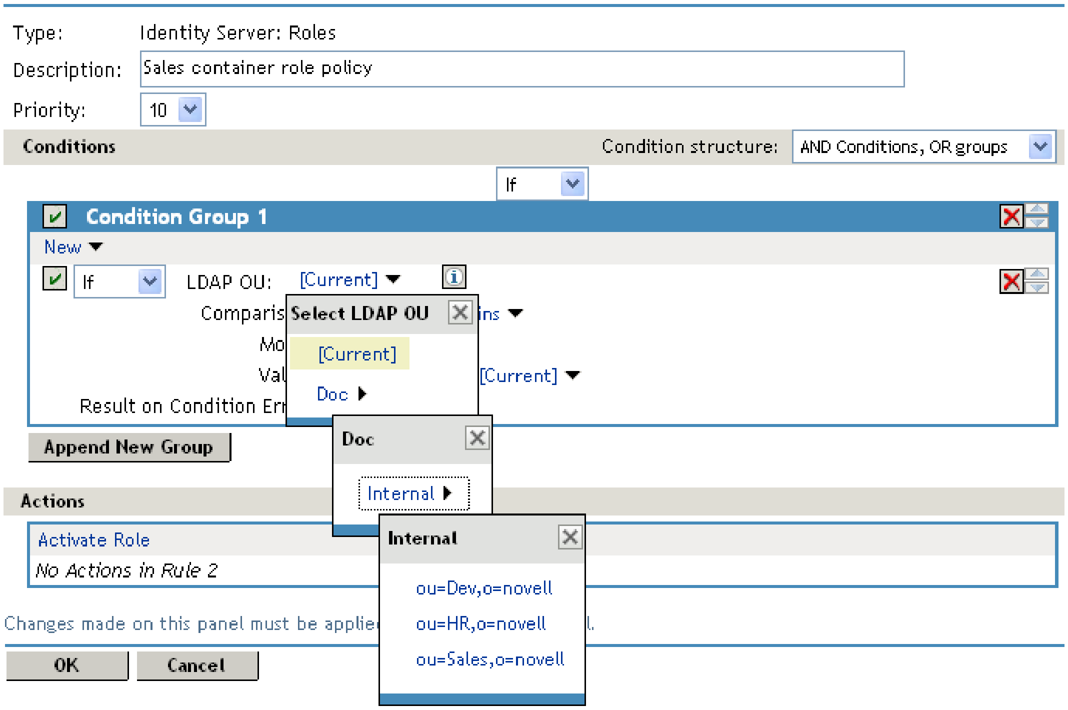
Figure 27-10 Using an eDirectory Tree for access control



Such a tree organization can be used to control access to resources. The following instructions explain how to create a Role policy for the users created under the Sales container.

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Click *New*, specify a name for the Role policy, select *Identity Server: Roles* for the type, then click *OK*.
3. In *Condition Group 1*, click *New*, and select *LDAP OU* > *[Identity Server Configuration]* > *[User Store]* > *[DN of the OU]*.

   The following example illustrates how to make these selections:



Type: Identity Server: Roles

Description: Sales container role policy

Priority: 10

Condition structure: AND Conditions, OR groups

LDAP OU: [Current]

Activate Role

No Actions in Rule 2

Changes made on this panel must be applie

Comparison: Select how you want the attribute values to be compared. For LDAP OU, select *Contains*.

Mode: Select *One Level* if all your users are created in ou=Sales. Select *Subtree* if your users are created in various containers under the ou=Sales container.

Value: Select *LDAP OU*, then select *[Current]*.

The DN of the authenticated user is compared with the value specified in LDAP OU. If the DN of the user contains the LDAP OU value, the user matches the condition. For example, if the DN of the user is cn=bsmith,ou=sales,o=novell and the LDAP OU value is ou=sales,o=novell, the user matches the condition. If you selected Subtree for the Mode, a user with the following DN also matches the condition: cn=djones,ou=provo,ou=sales,o=novell.

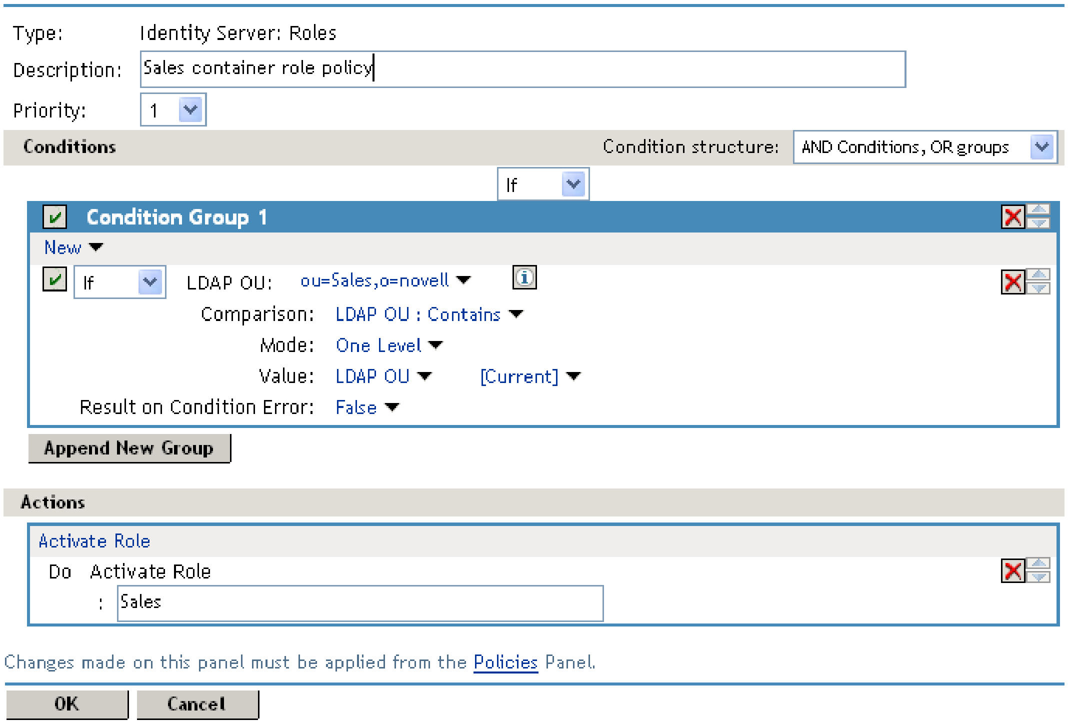
Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of Sales if the condition evaluates to *True*. If an error occurs, you do not want random users assigned the role of Sales. Therefore, for this rule, you need to select *False*.

4 In the *Actions* section, click *Activate Role*.

5 In the *Activate Role* box, type `Sales`, then click *OK*

The name you enter in the box is the role you want assigned to the users who match the condition.

Your rule should look similar to the following:



Type: Identity Server: Roles

Description: Sales container role policy

Priority: 1

Conditions

Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

☑ If LDAP OU: ou=Sales,o=novell

Comparison: LDAP OU : Contains

Mode: One Level

Value: LDAP OU [Current]

Result on Condition Error: False

Actions

Activate Role

Do Activate Role

: Sales

Changes made on this panel must be applied from the Policies Panel.

6 Click *OK* twice, then click *Apply Changes*.

7 To enable the role so that it can be used in Authorization and Identity Injection policies, click *Identity Servers > Edit > Roles*.

8 Select the check box by the name of the role, then click *Enable*.

9 Click *OK*.

10 To update the Identity Server, click *Servers > Update*.

You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see the following:

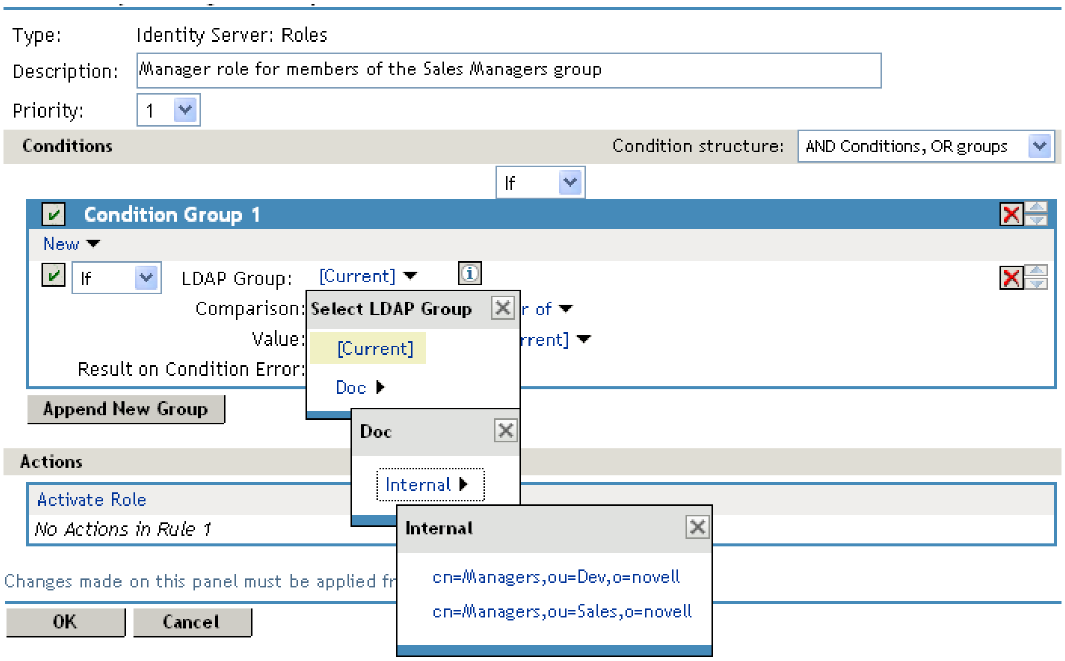
- Chapter 28, “Creating Authorization Policies,” on page 413
- Chapter 29, “Creating Identity Injection Policies,” on page 467

27.3.3 Creating a Role Using Role or Group Membership

If you have created an LDAP group and assigned users to the group, you can use group membership to assign a role to the user. For example, you might have created a first level managers group and made all your first level managers a member of this group. You would have other groups for your upper level managers. You can create a Role policy that assigns the user a role if the user is a member of a specific group. The Role policy can then be used in an Authorization or Identity Injection policy to protect a Web resource.

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Click *New*, specify a name for the Role policy, select *Identity Server: Roles* for the type, then click *OK*.
3. In *Condition Group 1*, click *New*, then select *LDAP Group*.
4. In *Condition Group 1*, select the conditions the user must meet:

   LDAP Group: Select the Identity Server Configuration, the user store, then the Group. The following figure illustrates this selection process.



Type: Identity Server: Roles
Description: Manager role for members of the Sales Managers group
Priority: 1
Condition structure: AND Conditions, OR groups
LDAP Group: [Current]

   Comparison: Select how you want the attribute values to be compared. For LDAP Group, select *Is Member of*.

   Value: Select *LDAP Group*, then select *[Current]*.

   The DN of the authenticated user is compared with the members of the LDAP Group. If the DN of the user matches one of the members, the user matches the condition.

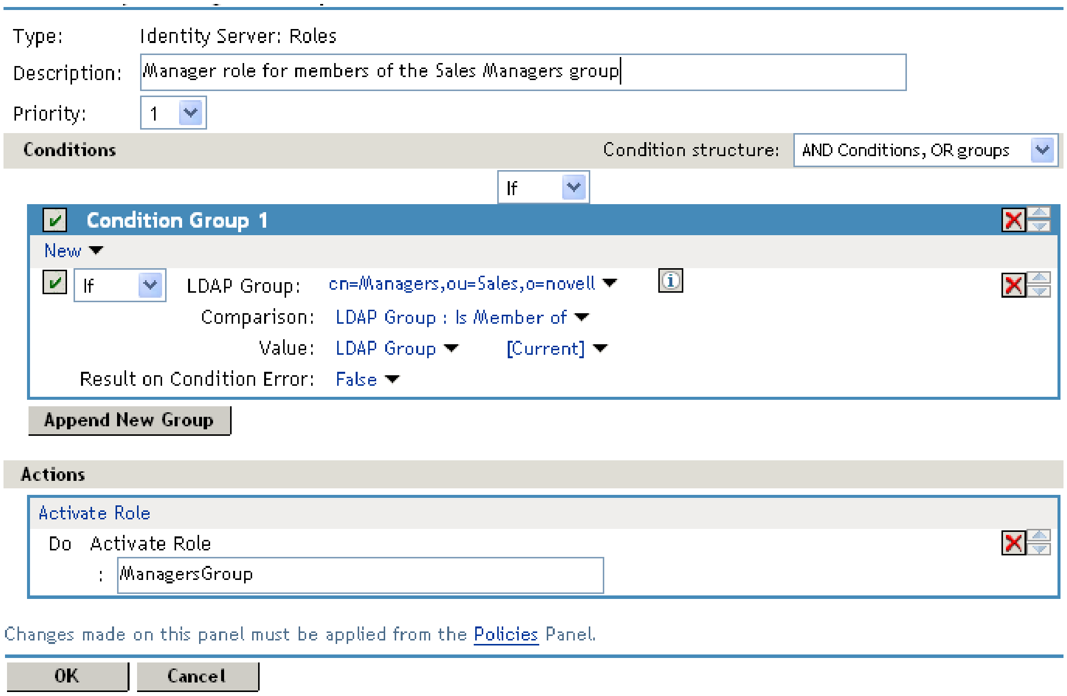
Result on Condition Error: This sets up the results that are returned if an error occurs while evaluating the condition (for example, the LDAP server goes down). This rule is set up to grant the user the role of ManagersGroup if the condition evaluates to *True*. If an error occurs, you do not want random users assigned the role of ManagersGroup. Therefore, for this rule, you need to select *False*.

5 In the *Actions* section, click *Activate Role*.

6 In the *Activate Role* box, type `ManagersGroup`, then click *OK*.

The name you enter in the box is the role you want assigned to the users who match the condition.

Your rule should look similar to the following:



Type: Identity Server: Roles

Description: Manager role for members of the Sales Managers group

Priority: 1

Conditions — Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

☑ If LDAP Group: cn=Managers,ou=Sales,o=novell

Comparison: LDAP Group : Is Member of

Value: LDAP Group [Current]

Result on Condition Error: False

Actions

Activate Role

Do Activate Role : ManagersGroup

Changes made on this panel must be applied from the Policies Panel.

7 Click *OK* twice, then click *Apply Changes*.

8 To enable the role so that it can be used in Authorization and Identity Injection policies, click *Identity Servers* > *Servers* > *Edit* > *Roles*.

9 Select the check box by the name of the role, then click *Enable*.

10 Click *OK*.

11 To update the Identity Server, click *Servers* > *Update Servers*.

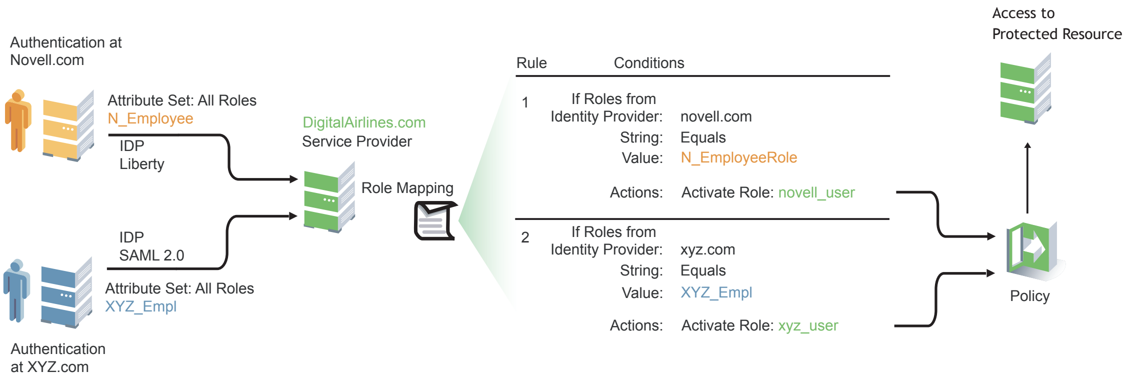
You can now use this role when creating Authorization and Identity Injection policies, which control access to protected Web resources. For more information, see:

- Chapter 28, "Creating Authorization Policies," on page 413
- Chapter 29, "Creating Identity Injection Policies," on page 467

27.4 Mapping Roles between Trusted Providers

The Identity Server can send roles in an authentication assertion. You can map these roles that are received from trusted providers to your own roles. Figure 27-11 illustrates this process.

Figure 27-11 *Role Mapping*



In this example, employees authenticate to identity providers novell.com (Liberty) or xyz.com (SAML 2.0). Each user is assigned to a role (such as N_EmployeeRole or XYZ_Empl, respectively). Attribute sets at each of the identity providers are configured to exchange the *All Roles* attribute with the trusted service provider, DigitalAirlines.com. DigitalAirlines.com consumes the authentication assertions, then maps the incoming roles to local roles. The mapped roles at DigitalAirlines.com can be used as evaluated conditions in authorization or J2EE policies, which can provide access to resources intended for the authenticated employees.

Prerequisites

- Configure trust between trusted providers, using the Liberty or SAML 2.0 protocol.

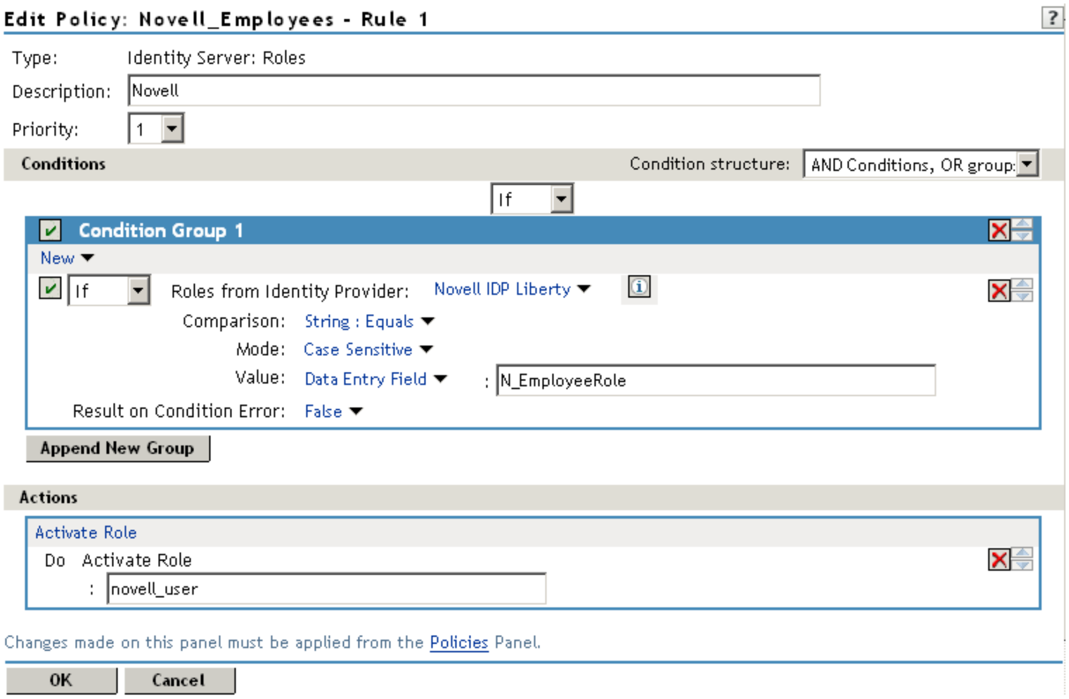
  You should be familiar with Chapter 9, "Configuring Trusted Providers," on page 139.
- Configure local authentication.

  You must create an external contract at the service provider that matches the contract of the identity provider. See Chapter 8, "Configuring Local Authentication," on page 89.
- Create an attribute set and select the local attribute *All Roles* in the set. This must be done at the identity provider and service provider.

  This attribute set is used to pass roles from an identity provider to an external service provider in authentication assertions. See Section 7.1, "Configuring Attribute Sets," on page 83.

The following procedure describes how the service provider configures this type of role policy for novell.com, mapping N_EmployeeRole to an Access Manager role:

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Click *New*, then specify a name for the Role policy.
3. Select *Identity Server: Roles* for the type, then click *OK*.
4. Configure the role policy as shown on the following page.



Edit Policy: Novell_Employees - Rule 1

Type: Identity Server: Roles

Description: Novell

Priority: 1

Conditions

Condition structure: AND Conditions, OR group:

If

☑ Condition Group 1

☑ If Roles from Identity Provider: Novell IDP Liberty

Comparison: String : Equals

Mode: Case Sensitive

Value: Data Entry Field : N_EmployeeRole

Result on Condition Error: False

Actions

Activate Role

Do Activate Role

: novell_user

Changes made on this panel must be applied from the Policies Panel.

5. In the *Conditions* section, click *New* > *Roles from Identity Provider*.
6. Select the trusted identity provider in the drop-down menu.
7. For *Comparison*, choose *String* > *Equals*.
8. Choose *Value* > *Data Entry Field*.
9. Type the name of the role used by the trusted identity provider.
10. Under the *Actions* section, click *Activate Role*.
11. Type the name of the role you want to activate at the trusted service provider.
12. Click *OK*.
13. On the Policies page, click *Apply Changes*.
14. To enable the role so that it can be used in Authorization and Identity Injection policies, click *Identity Servers* > *Servers* > *Edit* > *Roles*.
15. Select the check box by the name of the role, then click *Enable*.
16. Click *OK*.
17. To update the Identity Server, click *Servers* > *Update Servers*.

27.5 Enabling and Disabling Role Policies

In order for a role policy to function, you must enable it for the Identity Server configuration.

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Roles*.
2. Click the role policy’s check box, then click *Enable*.
3. To disable the role policy, click the role policy’s check box, then click *Disable*.
4. After enabling or disabling role policies, update the Identity Server configuration on the *Servers* tab.

27.6 Importing and Exporting Role Policies

You can import and export role policies in order to run them in other Identity Server configurations. When you import a role, ensure that you have enabled any Liberty profile that is referenced in the role policy, in order to correctly display the policy in the interface. However, the policy still evaluates if you have not enabled the profile.

You must also enable roles after importing them to an Identity Server configuration. See Section 27.5, "Enabling and Disabling Role Policies," on page 411.

When you export a role policy, the system saves it as a `.txt` file at the location of your choosing. After you import a role policy, you must update the Identity Server configuration.

To export a role policy:

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Select a policy, then click *Export*.
3. (Optional) Modify the name suggested for the file.
4. Click *OK*
5. Using the features of your browser, specify where the file is copied.

To import a role policy:

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Click *Import*, then browse to the location of the file.
3. Click *OK*.
4. When the policy appears in the list, click *Apply Changes*.

Creating Authorization Policies

28

Authorization policies are used when you want to protect a resource based on criteria other than authentication, and you want Access Manager to enforce the access restrictions. Authorization policies are enforced when a user requests data from a resource.

The Access Manager supports three types of Authorization policies:

- Access Gateway Authorization policies for protecting resources of the Access Gateway
- Web Authorization policies for protecting Java applications on a J2EE server
- Enterprise JavaBean Authorization policies for protecting the Enterprise JavaBeans of a J2EE application

The first step in creating an Authorization policy is determining the criteria for restricting access. The second step is translating those criteria into rules and conditions for a policy. This section describes the policy elements, but your resource and your security requirements determine which elements to use when creating the policy.

28.1 Designing an Authorization Policy

When creating an Authorization policy, you need to configure one or more rules. Each rule consists of two parts: (1) one or more conditions the user must meet and (2) the action to perform when the user meets the conditions or doesn’t meet the conditions. The action can be to either allow or deny access to the resource. This section describes how to use the following elements when creating a policy:

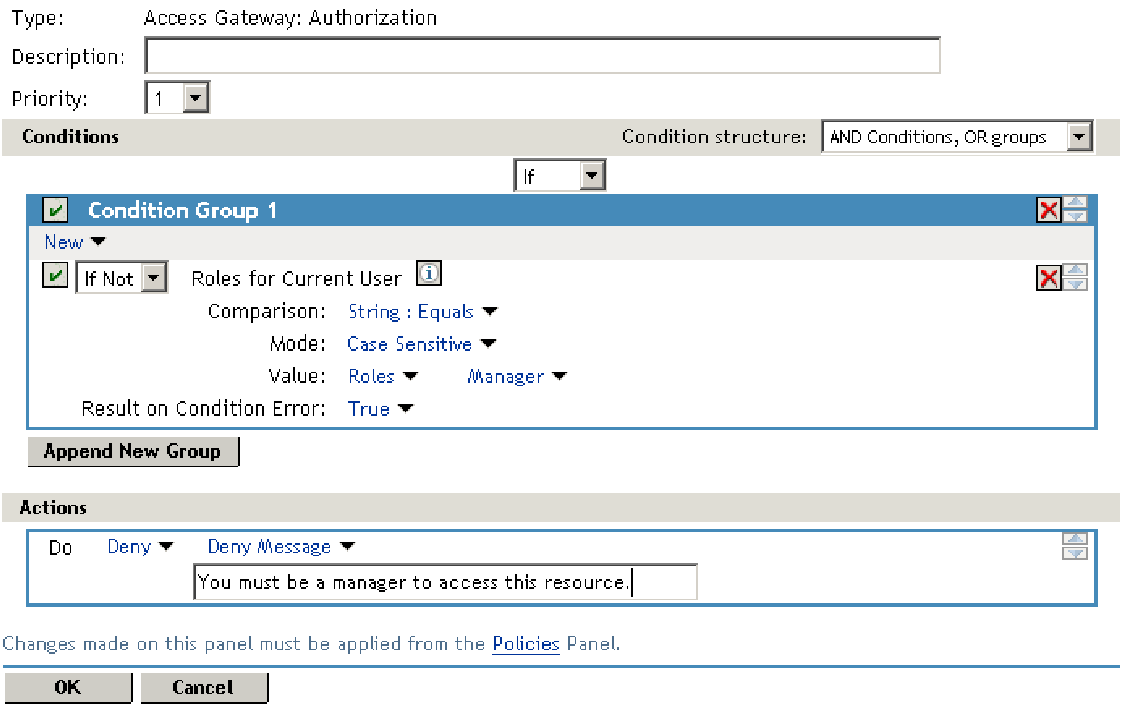
- Section 28.1.9, “Assigning Policies to Resources,” on page 420

28.1.1 Controlling Access with a Deny Rule and a Negative Condition

To deny access to the correct set of users, you need to know the characteristics of the users you don’t want accessing the resource, as well as the characteristics of the users you want accessing the resource.

Some very simple policies can be created using a Deny action. For example, suppose you have an application that you only want managers to access. If you have set up a role that assigns all managers to the Manager role, you can use this characteristic for an Authorization policy. Such a rule would be similar to the following:

Figure 28-1 *Simple Rule*



Type: Access Gateway: Authorization
Description:
Priority: 1
Conditions
Condition structure: AND Conditions, OR groups
If
☑ Condition Group 1
☑ If Not Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles Manager
Result on Condition Error: True
Actions
Do Deny Deny Message
You must be a manager to access this resource.
Changes made on this panel must be applied from the Policies Panel.

This rule evaluates the user, and if the user does not belong to the Manager role, the user matches the condition. The action for matching the condition is to deny access. The managers, who belong to the Manager role, do not match the condition and the Deny action is not applied to them.

The *Result on Condition Error* option is set to True. You don’t want an error to cause the policy to assume that the user is a manager. If an error occurs, you want the policy to assume that the user is not a manager, or in other words, matches the condition, and you want the Deny action applied.

28.1.2 Configuring the Result on Condition Error Option

The *Result on Condition Error* option allows you to specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either

False or *True*. You need to analyze the logic of your policy carefully, because if you set up this option incorrectly, error conditions can allow access to a resource. Consider the following:

- If your rule is a Permit rule and you do not want the action applied when an error occurs, select *False* for this option.
- If your rule is a Deny rule with an *If Not* condition and you want the action applied when an error occurs, select *True*.

28.1.3 Many Rules or Many Conditions

You can design your policy to have many rules with a single condition and action, or you can design your policy to have fewer rules, with each rule containing many conditions.

For example, suppose you have a resource that you don’t want users accessing on Monday, Wednesday, and Friday between 1:00 a.m. and 2:00 a.m. You could set up three rules, one for each day, or you could set up one rule with three conditions. If all the conditions have the same action (for example, deny access with the same reason), it is simpler to put them in the same rule. However, if you have a customized message to return for each day, you need to put them in separate rules.

Each rule contains the following:

- Zero or more conditions. A condition specifies how the request data is evaluated for a True or False match. Conditions are evaluated in the order in which they are listed.
- One or more condition groups. Conditions are placed in condition groups, which gives you the flexibility of creating a policy that allows the user to match the conditions in one group but not the conditions in the other condition groups. Or you can set up the condition groups to require that the user matches at least one condition in each condition group.
- An action, which grants access, denies access, or redirects the users.

Conditions, conditions groups, and the interaction among them allow you to create very simple rules (if A, then grant access) to very complex rules (if A, B, and C, but not D and E, then grant access).

28.1.4 Controlling Access with Multiple Conditions

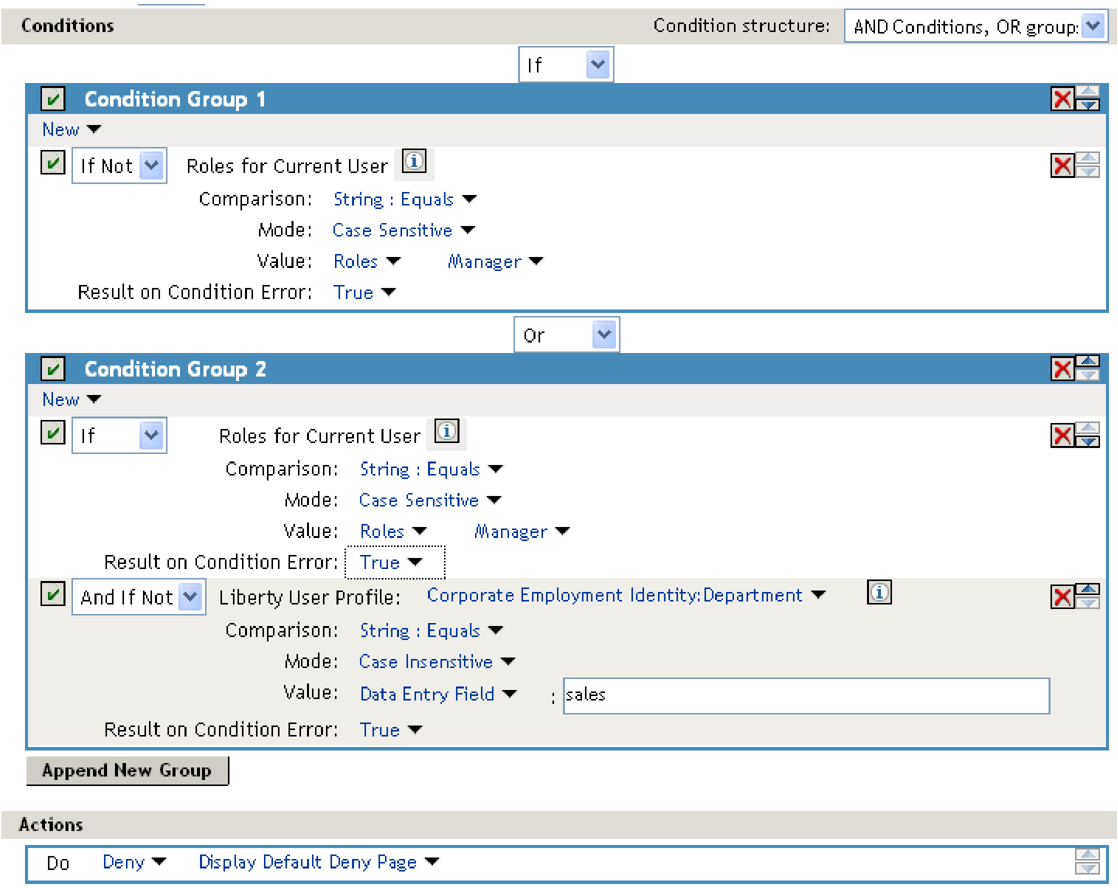
A policy requires multiple conditions when you have more than one required condition for granting access. For example, suppose you can easily identity your managers because they have all been assigned the role of Manager, and you have a resource that only the sales managers should access. Such a policy requires two conditions for granting access: the Manager role and membership in the sales department. For a Deny rule, the rule needs two condition groups:

- The first condition group matches all users who are not managers. This causes the Deny action to be applied.
- The second condition group matches the users who are managers but don’t belong to the sales department. Because they match both conditions, the Deny action is applied. For these two condition groups to work with this logic, the *Condition structure* is set to *AND Conditions, OR groups*.

The users who are managers and who belong to the sales department do not match either condition group. The Deny action is not applied, and they are allowed access.

Such a rule would look similar to the following:

Figure 28-2 A Rule with Two Condition Groups



Conditions

Condition structure: AND Conditions, OR group:

If

☑ Condition Group 1

☑ If Not Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles Manager
Result on Condition Error: True

Or

☑ Condition Group 2

☑ If Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles Manager
Result on Condition Error: True

☑ And If Not Liberty User Profile: Corporate Employment Identity:Department
Comparison: String : Equals
Mode: Case Insensitive
Value: Data Entry Field : sales
Result on Condition Error: True

Actions

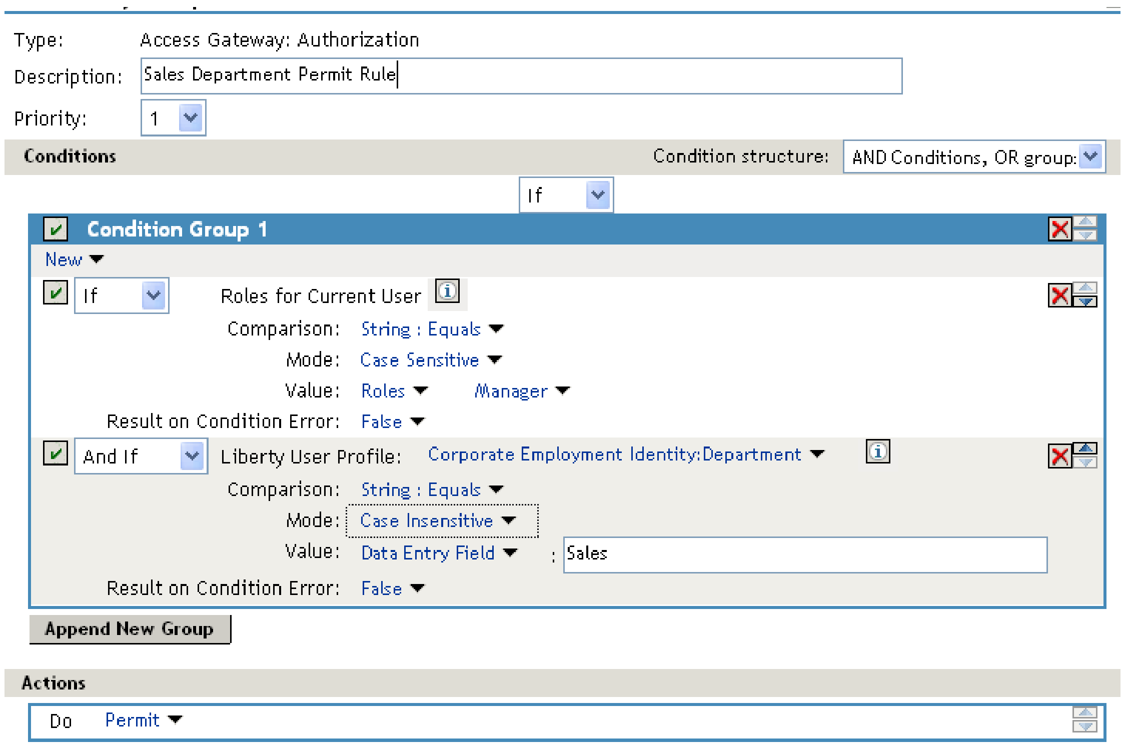
Do Deny Display Default Deny Page

This second condition group could be implemented as the second rule of the policy. If so, it should be set as a lower priority than the first rule. Because most systems would have more users than managers, the user rule would be used more frequently, so it should come first.

28.1.5 Using Permit Rules with a Deny Rule

You can also create policies that contain one or more Permit rules and then as the lowest priority rule in the policy, a Deny rule with no conditions. In such a policy, as soon as an allow match is processed, the rest of the rules are not processed and the user is granted access to the resource. The Deny rule is only processed if the user does not match one of the allow rules, and because all users match a rule with no conditions, the user is denied access to the resource. The first rule in such a policy for the sales application would look similar to the following.

Figure 28-3 *Rule 1 Granting Access*



Type: Access Gateway: Authorization

Description: Sales Department Permit Rule

Priority: 1

Conditions — Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

☑ If — Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles — Manager

Result on Condition Error: False

☑ And If — Liberty User Profile: Corporate Employment Identity:Department

Comparison: String : Equals

Mode: Case Insensitive

Value: Data Entry Field : Sales

Result on Condition Error: False

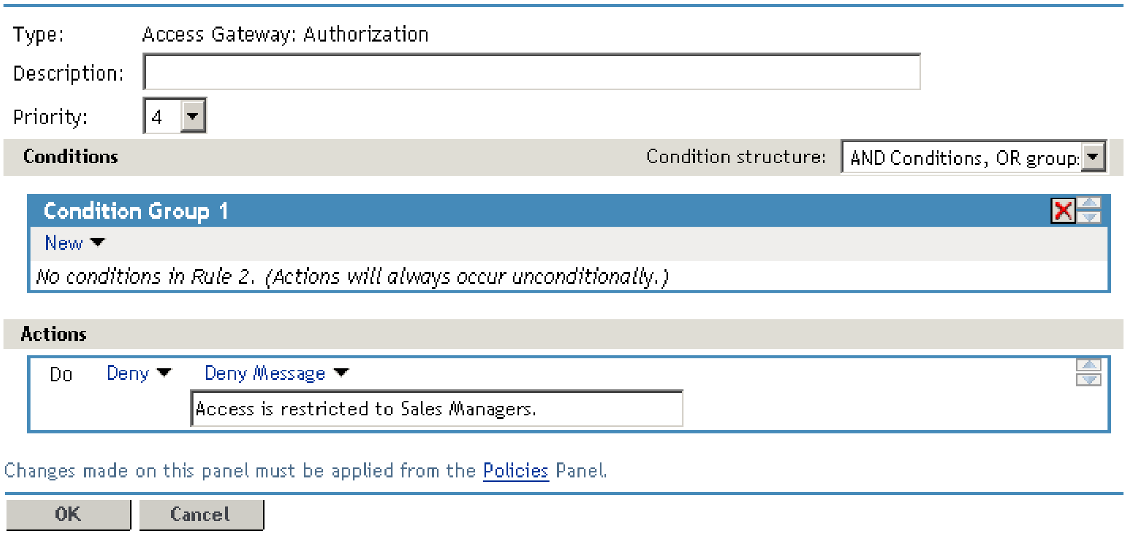
Actions

Do Permit

The conditions in Rule 1 are ANDed, which requires the user to match both conditions before they are granted access to the resource. The priority is set to 1, so this rule is the first rule that the Access Gateway processes. The J2EE authorization policies use the same logic.

The second rule would look similar to the following.

Figure 28-4 *Rule 2 Denying Access*



Type: Access Gateway: Authorization

Description:

Priority: 4

Conditions — Condition structure: AND Conditions, OR groups

Condition Group 1

No conditions in Rule 2. (Actions will always occur unconditionally.)

Actions

Do Deny — Deny Message

Access is restricted to Sales Managers.

Changes made on this panel must be applied from the Policies Panel.

Because this rule has no conditions, any user who does not match the first rule does match this rule and is denied access. The priority of this rule is set lower than the Permit rule so that the Permit rule is processed first.

28.1.6 Using Deny Rules with a General Permit Rule

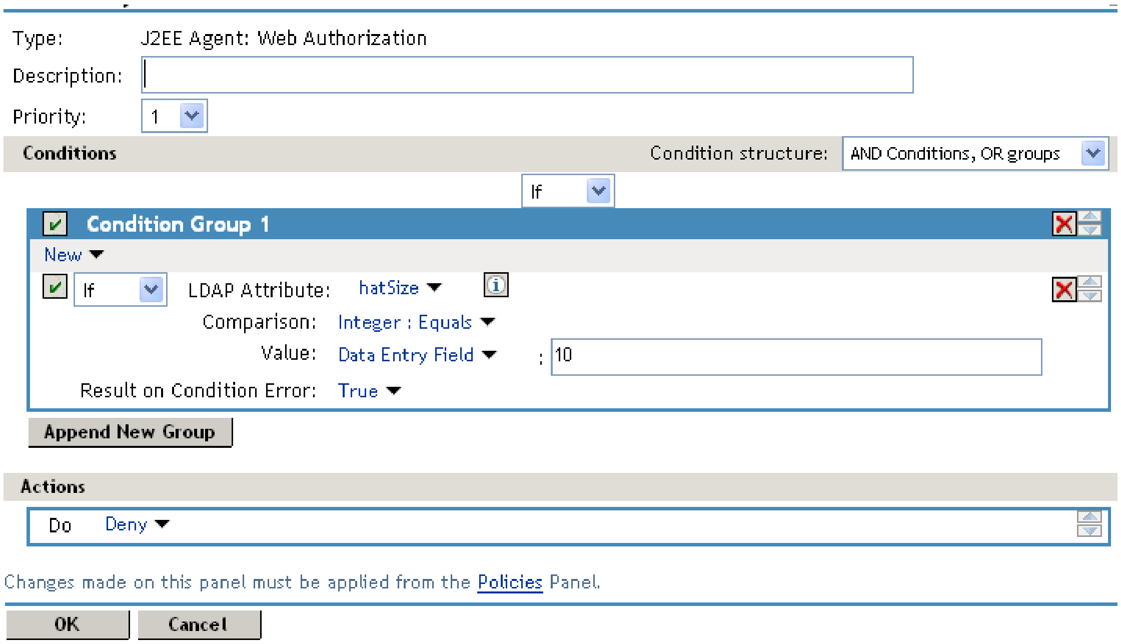
You can also create policies that contain one or more Deny rules and then as the lowest priority rule in the policy, a Permit rule with no conditions. In such a policy, as soon as a Deny rule matches a user, the rest of the rules are not processed and the user is denied access to the resource. The Permit rule is only processed if the user does not match one of the Deny rules. Because all users match a rule with no conditions, the user is allowed access to the resource.

The key to creating this type of policy is making sure all the Deny rules match the users you do not want accessing the resource and making sure that the *Result on Error Condition* option is set correctly.

For example, suppose one of the Deny rules uses an LDAP attribute for the condition and that the attribute is a hatSize attribute. Some of your users do not have a hatSize attribute, so when they access the resource, the comparison generates an error. If *Result on Error Condition* option is set to False, the action (Deny) is not applied, and the next rule in the policy is processed. If that rule is the general Permit rule, then they are allowed access to the resource because they experienced an error. To prevent this behavior, you need to set the *Result on Error Condition* option to True, so that the Deny action is applied. Your rule then denies access to everyone whose hatSize attribute matches the specified value and everyone who does not have the attribute.

The Deny rule for such a policy would look similar to the following:

Figure 28-5 *Deny Rule Configured for Error Conditions*



Type: J2EE Agent: Web Authorization
Description:
Priority: 1
Condition structure: AND Conditions, OR groups
If
Condition Group 1
If LDAP Attribute: hatSize
Comparison: Integer : Equals
Value: Data Entry Field : 10
Result on Condition Error: True
Do Deny
Changes made on this panel must be applied from the Policies Panel.

For most people, Deny rules are harder to write than Permit rules. You not only need to carefully configure the *Result on Condition Error* option, you must also carefully consider the consequences of the condition not matching a user. When a user doesn't match the condition, the Action is not applied and the next rule in the policy is evaluated. For example, suppose the URL condition is set to the compare the following value:

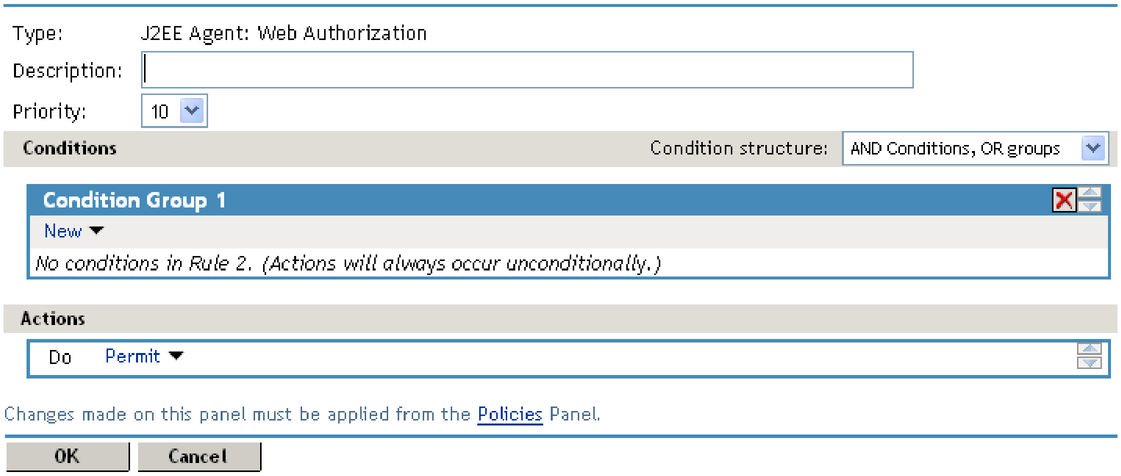
`http://sales.provo.novell.com/meetings/?`

If the URL in the request is `http://sales.provo.novell.com/meetings/january`, the user does not match the condition, because the ? applies only to the files in the `meetings` directory and not to the subdirectories. The Action is not applied, and the next rule or policy is evaluated. Consider the following possibilities:

- If you want the condition to match all files and subdirectories, you need to change the ? wildcard to the * wildcard.
- If you want the condition to allow access to the files in the `/meetings` directory but deny access to the subdirectories, you need to negate the condition so it evaluates as follows: if the URL is not a request for the `/meetings/?` directory, deny access. If you select this type of condition, you need to set the *Result on Condition Error* option to True. If the comparison returns an error and there is the possibility that the request is for a subdirectory, you want the user to be denied access.

The general Permit rule for a Deny policy would look similar to the following:

Figure 28-6 *General Permit Rule*



Type: J2EE Agent: Web Authorization

Description:

Priority: 10

Conditions — Condition structure: AND Conditions, OR groups

Condition Group 1

New

No conditions in Rule 2. (Actions will always occur unconditionally.)

Actions

Do Permit

Changes made on this panel must be applied from the Policies Panel.

NOTE: This type of policy is not recommended for WebSphere applications protected by the J2EE Agent. WebSphere, even when the user is logged in, always first uses the anonymous user to access resources and switches to the actual username only when the anonymous user is denied. If the policy uses conditions that require information that is available only if the user is authenticated, this type of policy produces unexpected results.

28.1.7 Public Policies

You can create public authorization policies, which are policies that apply to everyone, by leaving the *Condition* section empty. In the *Action* section, you specify either to deny or to permit access to the resource. Then you assign the policy to the protected resource.

28.1.8 General Design Principles

When designing a policy, remember the following principles:

- Logged-in users are allowed access to a protected resource unless the policy denies access.
- Priority determines the order in which rules are applied.
- The Conditions section of the rule must evaluate to True in order for the Action section to be applied. If the Condition section evaluates to False, the Action section is ignored and the policy moves to the next rule. If another rule does not exist, the user is granted access to the resource.
- Rules are only processed until a user matches the conditions in a rule and its action is applied. If a user matches the first rule in a policy, that action is applied, and the rest of the rules in the policy are ignored.
- If two rules have the same priority, Deny rules are applied before Permit rules.
- After you have designed your policy, created it, and assigned it to a resource, you need to test the policy. You need to log in as the type of user who should be granted access, as the type of user who should not be granted access, and as user who generates an error on condition evaluation.

28.1.9 Assigning Policies to Resources

For information on how to assign the policy to a resource, see the following:

- For an Access Gateway policy, see Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210.
- For a Web Authorization policy, see “Assigning a Web Authorization Policy to the Resource” in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*.
- For a Enterprise JavaBean Authorization policy, see “Assigning an Enterprise JavaBeans Authorization Policy to a Resource” in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*.

28.2 Creating Access Gateway Authorization Policies

An Authorization policy specifies conditions that a user must meet in order to access a resource. The Access Gateway enforces these conditions. The policy specifies the criteria a user must meet to either allow access or deny access.

To create an Access Gateway Authorization policy:

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, then select *Access Gateway: Authorization* for the type of policy.
3. Fill in the following fields:

   Description: (Optional) Describe the purpose of this rule.

   Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and 10 is the lowest. If two rules have the same priority, a Deny rule is applied before a Permit rule.

4 In the *Condition Group 1* section, click *New*, then select one of the following:

- **Authentication Contract:** Allows you to control access based on the contract the user used for login. For configuration information, see Section 28.5.1, "Authentication Contract Condition," on page 426.
- **Client IP:** Allows you to control access based on the IP address of the client making the request. For configuration information, see Section 28.5.2, "Client IP Condition," on page 428.
- **Credential Profile:** Allows you to control access based on the credentials the user specified during authentication. For configuration information, see Section 28.5.3, "Credential Profile Condition," on page 429.
- **Current Date:** Allows you to control access based on the date of the request. For more information, see Section 28.5.4, "Current Date Condition," on page 431.
- **Current Day of Week:** Allows you to control access based on the day the request is made. For configuration information, see Current Day of Week Condition.
- **Current Day of Month:** Allows you to control access based on the month the request is made. For configuration information, see Section 28.5.6, "Current Day of Month Condition," on page 433.
- **Current Time of Day:** Allows you to control access based on the time the request was made. For configuration information, see Section 28.5.7, "Current Time of Day Condition," on page 434.
- **HTTP Request Method:** Allows you to control access based on the request method. For configuration information, see Section 28.5.8, "HTTP Request Method Condition," on page 435.
- **LDAP Attribute:** Allows you to control access based on the value of an LDAP attribute. For configuration information, see Section 28.5.9, "LDAP Attribute Condition," on page 436.
- **LDAP OU:** Allows you to control access based on the value of an LDAP organizational unit. For configuration information, see Section 28.5.10, "LDAP OU Condition," on page 437.
- **Liberty User Profile:** Allows you to control access based on the value of a profile attribute. For configuration information, see Section 28.5.11, "Liberty User Profile Condition," on page 437.
- **Roles for Current User:** Allows you to control access based on the roles a user has been assigned. For configuration information, see Section 28.5.12, "Roles for Current User Condition," on page 438.
- **URL:** Allows you to control access based on the URL in the request. For configuration information, see Section 28.5.13, "URL Condition," on page 439.
- **URL Scheme:** Allows you to control access based on the scheme in the URL of the request (for example, http or https). For configuration information, see Section 28.5.14, "URL Scheme Condition," on page 441.
- **URL Host:** Allows you to control access based on the hostname in the URL of the request. For configuration information, see Section 28.5.15, "URL Host Condition," on page 442.
- **URL Path:** Allows you to control access based on the path in the URL of the request. For configuration information, see Section 28.5.16, "URL Path Condition," on page 443.

- **URL File Name:** Allows you to control access based on the filename in the URL of the request. For configuration information, see Section 28.5.17, "URL File Name Condition," on page 445.
- **URL File Extension:** Allows you to control access based on the file extension in the URL of the request. For configuration information, see Section 28.5.18, "URL File Extension Condition," on page 446.
- **X-Forwarded-For IP:** Allows you to control access based on the value in the X-Forwarded-For IP header of the HTTP request. For configuration information, see Section 28.5.19, "X-Forward-For IP Condition," on page 447.

5. To add multiple conditions to the same rule, either add a condition to the same condition group or create a new condition group. For information on how conditions and condition groups interact with each other, see Section 28.7, "Using Multiple Conditions," on page 464.
6. In the *Actions* section, select either *Permit*, *Deny*, or *Redirect*.

   If you select *Redirect*, specify the URL to which you want users redirected when they meet the conditions of this policy.

   If you select *Deny*, select one of the following:

   - **Display Default Deny Page:** Displays a generic message, indicating that users have insufficient rights to access the resource.
   - **Deny Message:** Allows you to provide a customized message that is displayed to users who are denied access. This message can be plain text or text with HTML tags.
   - **Redirect to URL:** Allows you to specify a URL to which users are redirected when they are denied access. For example:

     `http://www.novell.com`
7. To save the rule, click *OK*.
8. To add another rule, click *New* or to save the policy, click *OK*, then click *Apply Changes*.
9. For information on how to assign the policy to a protected resource, see Section 13.4.2, "Assigning an Authorization Policy to a Protected Resource," on page 210.

28.3 Creating Web Authorization Policies for J2EE Agents

A Web Authorization policy specifies conditions that a user must meet in order to access a resource on a J2EE server. The Web Authorization policy specifies the criteria a user must meet to either allow access or deny access. For example, if you create a Sales role and assign it to the users, the role can be used to allow access to the sales applications and to deny access to resource management applications. For information about designing a policy, see Section 28.1, "Designing an Authorization Policy," on page 413.

To create a Web Authorization policy:

1. In the Administration Console, click *Access Manager > Policies > New*.
2. Specify a name for the policy, select *J2EE Agent: Web Authorization* as the type, then click *OK*.
3. Fill in the following fields:

   Description: (Optional) Specify a description for the rule.

Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and 10 is the lowest. If two rules have the same priority, a Deny rule is applied before a Permit rule.

4 In the *Condition Group 1* section, click *New*, then select one of the following:

- **Client IP Address:** Allows you to control access based on the IP address of the client making the request. For configuration information, see Section 28.5.2, "Client IP Condition," on page 428.
- **Credential Profile:** Allows you to control access based on the credentials the user specified during authentication. For configuration information, see Section 28.5.3, "Credential Profile Condition," on page 429.
- **Current Date:** Allows you to control access based on the date of the request. For more information, see Section 28.5.4, "Current Date Condition," on page 431.
- **Current Day of Week:** Allows you to control access based on the day the request is made. For configuration information, see Section 28.5.5, "Current Day of Week Condition," on page 432.
- **Current Day of Month:** Allows you to control access based on the month the request is made. For configuration information, see Section 28.5.6, "Current Day of Month Condition," on page 433.
- **Current Time of Day:** Allows you to control access based on the time the request was made. For configuration information, see Section 28.5.7, "Current Time of Day Condition," on page 434.
- **HTTP Request Method:** Allows you to control access based on the request method. For configuration information, see Section 28.5.8, "HTTP Request Method Condition," on page 435.
- **LDAP Attribute:** Allows you to control access based on the value of an LDAP attribute. For configuration information, see Section 28.5.9, "LDAP Attribute Condition," on page 436.
- **Liberty User Profile:** Allows you to control access based on the value of a profile attribute. For configuration information, see Section 28.5.11, "Liberty User Profile Condition," on page 437.
- **Roles for Current User:** Allows you to control access based on the roles a user has been assigned. For configuration information, see Section 28.5.12, "Roles for Current User Condition," on page 438.
- **URL:** Allows you to control access based on the URL in the request. For configuration information, see Section 28.5.13, "URL Condition," on page 439.
- **URL Scheme:** Allows you to control access based on the scheme in the URL of the request (for example, http or https). For configuration information, see Section 28.5.14, "URL Scheme Condition," on page 441.
- **URL Host:** Allows you to control access based on the hostname in the URL of the request. For configuration information, see Section 28.5.15, "URL Host Condition," on page 442.
- **URL Path:** Allows you to control access based on the path in the URL of the request. For configuration information, see Section 28.5.16, "URL Path Condition," on page 443.
- **URL File Name:** Allows you to control access based on the filename in the URL of the request. For configuration information, see Section 28.5.17, "URL File Name Condition," on page 445.

- **URL File Extension:** Allows you to control access based on the file extension in the URL of the request. For configuration information, see Section 28.5.18, “URL File Extension Condition,” on page 446.
- **X-Forwarded-For IP:** Allows you to control access based on the value in the X-Forwarded-For IP header of the HTTP request. For configuration information, see Section 28.5.19, “X-Forward-For IP Condition,” on page 447.

5. To add multiple conditions to the same rule, either add a condition to the same condition group or create a new condition group. For information on how conditions and condition groups interact with each other, see Section 28.7, “Using Multiple Conditions,” on page 464.
6. In the *Actions* section, select either *Permit* or *Deny*.
7. To save the rule, click *OK* twice, then click *Apply Changes*.
8. Assign the policy to a Web resource. See “Assigning a Web Authorization Policy to the Resource” in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*

28.4 Creating Enterprise JavaBean Authorization Policies for J2EE Agents

An Enterprise JavaBean (EJB*) Authorization policy allows you to protect the entire bean or specific interfaces or methods. For information about designing a policy, see Section 28.1, “Designing an Authorization Policy,” on page 413.

To create an EJB Authorization policy:

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *J2EE Agent: EJB Authorization* as the type, then click *OK*.
3. Fill in the following fields:

   Description: (Optional) Specify a description for the rule.

   Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and 10 is the lowest. If two rules have the same priority, a Deny rule is applied before a Permit rule.
4. In the *Condition Group 1* section, click *New*, then select one of the following:
   - **Credential Profile:** Allows you to control access based on the credentials the user specified during authentication. For configuration information, see Section 28.5.3, “Credential Profile Condition,” on page 429.
   - **Current Date:** Allows you to control access based on the date of the request. For more information, see Section 28.5.4, “Current Date Condition,” on page 431.
   - **Current Day of Week:** Allows you to control access based on the day the request is made. For configuration information, see Section 28.5.5, “Current Day of Week Condition,” on page 432.
   - **Current Day of Month:** Allows you to control access based on the month the request is made. For configuration information, see Section 28.5.6, “Current Day of Month Condition,” on page 433.
   - **Current Time of Day:** Allows you to control access based on the time the request was made. For configuration information, see Section 28.5.7, “Current Time of Day Condition,” on page 434.

- **LDAP Attribute:** Allows you to control access based on the value of an LDAP attribute. For configuration information, see Section 28.5.9, "LDAP Attribute Condition," on page 436.
- **Liberty User Profile:** Allows you to control access based on the value of a profile attribute. For configuration information, see Section 28.5.11, "Liberty User Profile Condition," on page 437.
- **Roles for Current User:** Allows you to control access based on the roles a user has been assigned. For configuration information, see Section 28.5.12, "Roles for Current User Condition," on page 438.

5 To add multiple conditions to the same rule, either add a condition to the same condition group or create a new condition group. For information on how conditions and condition groups interact with each other, see Section 28.7, "Using Multiple Conditions," on page 464.

6 In the *Actions* section, select either *Permit* or *Deny*.

7 To save the rule, click *OK*, then click *Apply Changes*.

8 Assign the policy to an EJB resource. See "Assigning an Enterprise JavaBeans Authorization Policy to a Resource" in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*

28.5 Conditions

This section describes the possible conditions for an Authorization policy. Some conditions can be set up so that the current values in the request are compared against static values (A to B), or you can compare static values to current values in the request (B to A). Within one policy, you should probably decide which direction to set up the comparisons and remain consistent unless there is a compelling reason to switch the direction for a particular condition.

For example, suppose you set up a rule to allow access to a resource only during the weekdays (Monday through Friday). You set up four of these conditions to compare if the date when the request is made matches with Monday, Tuesday, Wednesday, or Thursday. You set up the fifth condition to compare whether Friday matches the date when the request is made. This works, but maintaining this policy is more difficult because each new policy manager will ponder about the Friday condition and wonder why it is configured differently.

Many conditions, when used as the sole condition of a rule, do not make very useful rules. For example, you can create a rule that grants access if the user specifies a specific URL in the request. Such a rule has limited application. But a rule that requires that the request contain a specific URL and that the user have a specific role has greater application because it can be used to limit access to the URL based on the user's role. For information about how conditions can be ANDed or ORed together or placed in different condition groups, see Section 28.7, "Using Multiple Conditions," on page 464.

Authorization policies use the following conditions:

- Section 28.5.1, "Authentication Contract Condition," on page 426
- Section 28.5.2, "Client IP Condition," on page 428
- Section 28.5.3, "Credential Profile Condition," on page 429
- Section 28.5.4, "Current Date Condition," on page 431
- Section 28.5.5, "Current Day of Week Condition," on page 432
- Section 28.5.6, "Current Day of Month Condition," on page 433

- Section 28.5.7, "Current Time of Day Condition," on page 434
- Section 28.5.8, "HTTP Request Method Condition," on page 435
- Section 28.5.9, "LDAP Attribute Condition," on page 436
- Section 28.5.10, "LDAP OU Condition," on page 437
- Section 28.5.11, "Liberty User Profile Condition," on page 437
- Section 28.5.12, "Roles for Current User Condition," on page 438
- Section 28.5.13, "URL Condition," on page 439
- Section 28.5.14, "URL Scheme Condition," on page 441
- Section 28.5.15, "URL Host Condition," on page 442
- Section 28.5.16, "URL Path Condition," on page 443
- Section 28.5.17, "URL File Name Condition," on page 445
- Section 28.5.18, "URL File Extension Condition," on page 446
- Section 28.5.19, "X-Forward-For IP Condition," on page 447

For the specific policies they can be used in, see the following:

- Section 28.2, "Creating Access Gateway Authorization Policies," on page 420
- Section 28.3, "Creating Web Authorization Policies for J2EE Agents," on page 422
- Section 28.4, "Creating Enterprise JavaBean Authorization Policies for J2EE Agents," on page 424

28.5.1 Authentication Contract Condition

The *Authentication Contract* condition matches the contract the user logged in with to the contract specified in this condition. The Identity Server has the following default contracts:

Name	URI
Name/Password - Basic	basic/name/password/uri
Name/Password - Form	name/password/uri
Secure Name/Password - Basic	secure/basic/name/password/uri
Secure Name/Password - Form	secure/name/password/uri

To configure other contracts for your system, click *Identity Servers* > *Edit* > *Local* > *Contracts*.

To specify an *Authentication Contract* condition, fill in the following fields:

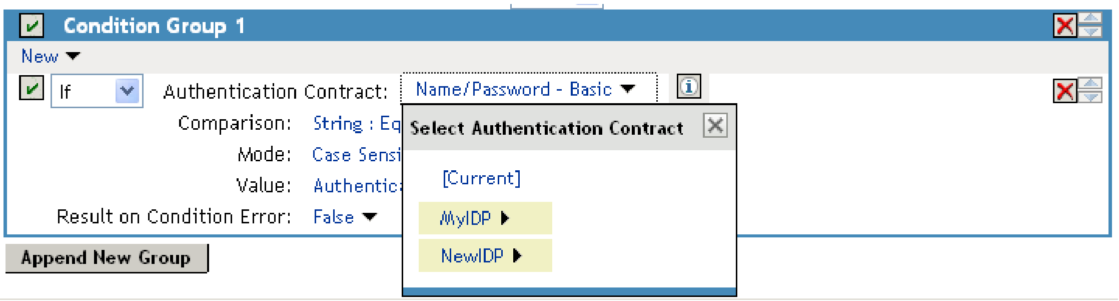
Authentication Contract: To compare the contract that the user used with a static value, select *Current*. To compare a static value with what the user used, select a contract from the list.

If you have created more than one Identity Server configuration, select the configuration that corresponds to the configuration your Access Gateway is configured to trust, then select the contract. The name of the contract is displayed. When you select this name, the configurations that contain a definition for this contract are highlighted.

If you select a contract that is defined on only one of your configurations, be aware that you must change this policy when you change configurations. If you select a contract that is defined in all your configurations, this policy requires no modifications and continues to function when you change configurations.

For example, the following policy has selected Name/Password - Basic as the contract.

Figure 28-7 *An Authentication Contract Defined by Multiple Identity Server Configurations*



If

Authentication Contract: Name/Password - Basic

Comparison: String : Eq

Mode: Case Sensi

Value: Authentic

Result on Condition Error: False

Two Identity Server configurations have been defined (MyIDP and New IDP). Both configurations are highlighted because Name/Password - Basic is a contract that is automatically defined for all Identity Server configurations. Because it is defined on both configurations, this policy’s function is the same, regardless of which configuration is selected as the trusted configuration.

Comparison: Specify how the contract is compared to the data in the *Value* field. Select either a string comparison or a regular expression:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the *Authentication Contract* value must begin with the letters specified in the *Value* field.
  - **Ends with:** Indicates that the *Authentication Contract* value must end with the letters specified in the *Value* field.
  - **Contains Substring:** Indicates that the *Authentication Contract* value must contain the letters, in the same sequence, as specified in the *Value* field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode

Unix Lines

Value: Specify the value you want to compare with the *Authentication Contract* value. If you select a static value for the *Authentication Contract* value, select *Authentication Contract* and *Current*. If you select *Current* for the *Authentication Contract* value, select *Authentication Contract*, then select the name of a contract.

Other value types are possible if you selected *Current* for the *Authentication Contract* value. For example:

- You can select *Data Entry Field*. The value specified in the text box must be the URI of the contract for the conditions to match. For a list of these values, click *Access Manager > Identity Servers > Edit > Local > Contracts*.
- If you have defined a Liberty User Profile attribute for the URI of authentication contracts, you can select *Liberty User Profile* and your defined attribute.
- If you have defined an LDAP attribute for the URI of the authentication contracts, you can select *LDAP Attribute* and your defined attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.2 Client IP Condition

The *Client IP* condition allows you to use the IP address of the user making the request to determine whether the user is allowed access to a resource.

Fill in the following fields:

Comparison: Specify how the client IP address is compared to the data in the *Value* field. Select either an IP comparison or a regular expression:

- **Comparison: IP:** Specifies that you want the values compared as IP addresses. Select one of the following:
  - **Equals:** Allows you to specify an IP address that the client must match. You can specify more than one.
  - **In Range:** Allows you to specify a range of IP addresses that the client's address must fall within. You can specify more than one range.
  - **In Subnet:** Allows you to specify the subnet that the client's address must belong to. You can specify more than one subnet.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode

Unix Lines

Value: Select *Data Entry Field* and specify a value appropriate for your comparison type. Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line. Use the *Add* button to add values one at a time. For example:

Comparison Type	Value

Equals	`10.10.10.10`
	`10.10.10.11`

In Range	`10.10.10.10 - 10.10.10.100`
	`10.10.20.10 - 10.10.20.100`

In Subnet	`10.10.10.12 / 22`
	`10.10.20.30 / 22`

Other values types are possible. For example, if your user store contains an LDAP attribute with the IP address of your users, you could select to compare the client's current IP address with the stored value by using an LDAP attribute or a Liberty User Profile value.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.3 Credential Profile Condition

The *Credential Profile* condition allows you to control access based on the credentials the user entered when authenticating to the system.

To set up the matching for this condition, fill in the following fields:

Credential Profile: Specify the type of credential your users are using for authentication. Select one of the following:

- **LDAP Credentials:** If you prompt the user for a user name, select this option, then select *LDAP User Name* (the cn of the user), *LDAP User DN* (the fully distinguished name of the user), or *LDAP Password*.

  The default contracts assign the cn attribute to the Credential Profile. If you create your own authentication contract, you can assign a different attribute to the Credential Profile.

  If your user store is an Active Directory server, you need to be aware that the cn attribute is used even though the user login is chosen from the SAMAccountName attribute. If you want to use the SAMAccountName attribute in the Credential Profile, you need to create your own authentication contract.
- **X509 Credentials:** If you prompt the user for a certificate, select this option, then one of the following:
  - **X509 Public Certificate Subject:** Retrieves the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Retrieves the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.

    - **X509 Public Certificate:** Retrieves the entire certificate, Base64 encoded.
    - **X509 Serial Number:** Retrieves the serial number of the certificate.
- **SAML Credential:** If your users authenticate using a SAML assertion, select this option.

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
    - **Equals:** Indicates that the values must match, letter for letter.
    - **Starts with:** Indicates that the *Credential Profile* value must begin with the letters specified in the *Value* field.
    - **Ends with:** Indicates that the *Credential Profile* value must end with the letters specified in the *Value* field.
    - **Contains Substring:** Indicates that the *Credential Profile* value must contain the letters, in the same sequence, as specified in the *Value* field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the second value for the comparison. Select one of the following data types:

- **LDAP Attribute:** If you have an LDAP attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Liberty User Profile:** If you have a Liberty User Profile attribute that corresponds to the Credential Profile you have specified, select this option and the attribute.
- **Data Entry Field:** Specify the string you want matched. Be aware of the following requirements:
    - If you selected *LDAP User DN* as the credential, you need to specify the DN of the user in the *Value* text box. If the comparison type is set to *Contains Substring*, you can match a group of users by specifying a common object that is part of their DNs, for example ou=sales.
    - If you selected *X509 Public Certificate Subject* as the credential, you need to specify all elements of the Subject Name of the certificate in the *Value* text box. Separate the elements with a comma and a space, for example, o=novell, ou=sales. If the comparison

type is set to *Contains Substring*, you can match a group of certificates by specifying a name that is part of their Subject Name, for example ou=sales.

Other values are possible. Your policy requirements determine whether they are useful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.4 Current Date Condition

The *Current Date* condition allows you to use the date to determine whether the user is allowed access to a resource.

Fill in the following fields:

Comparison: Specify how the current date is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: Date:** Specifies that you want the values compared as dates. Select one of the following date operators:
    - **Equals:** Requires that the current date must equal the specified value.
    - **Greater Than:** Requires that the current date be after the specified value.
    - **Greater Than or Equal to:** Requires that the current date be after or equal to the specified value.
    - **Less Than:** Requires that the current date be before the specified value.
    - **Less Than or Equal to:** Requires that the current date be before or equal to the specified value.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Date Format: If you selected a date comparison, specify the format of the *Value* field. Select one of the following formats:

- **D/M/Y** = 1/Jul/2007 or 1/7/2007
- **D-M-Y** = 1-Jul-2007 or 1-7-2007
- **D.M.Y** = 1.Jul.2007 or 1.7.2007
- **M/D/Y** = Jul/1/2007 or 7/1/2007
- **M-D-Y** = Jul-1-2007 or 7-1-2007
- **M.D.Y** = Jul.1.2007 or 7.1.2007
- **YYYY-MM-DD** = 2007-07-01
- **YYYY.MM.DD** = 2007.07.01

D specifies a number from 1 to 31. *M* specifies a number from 1 to 12 or the name of the month in three letters (Sep) or complete (September). *Y* specifies the year in a four digit format.

Mode: If you selected a regular expression comparison, select the mode for the comparison. Select one or more of the following:

Canonical Equivalence
Case Insensitive
Comments
Dot All
Multi-Line
Unicode
Unix Lines

Value: Specify the second value for the comparison. If you select *Data Entry Field* as the value type, specify the date in the format you select in the *Date Format* field.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to the date, you can use this option and select your attribute. The *Date Format* field does not apply to these value types.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.5 Current Day of Week Condition

The *Current Day of Week* condition allows you to restrict access based on which day of the week the request is made. Fill in the following fields:

Current Day of Week: Select the name of the day from the list. To compare the day specified in the current request with a static value, select *Current*. To compare a static value with the day specified in the current request, select the name of a day from the list.

Comparison: Specify how the current day of the week is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: Day of Week:** Specifies that you want the values compared as a day of the week. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the client must match.
  - **In Range:** Allows you to specify a range of days that the client's request must fall within, for example, Monday to Friday.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the second value for the comparison. If you select *Current* for the *Current Day of Week* field, you need to specify a static value. If you select a static value for the *Current Day of the*

Week field, you need to select *Current* for the *Value* field. If you select *Data Entry Field* as the value type, days of the week are specified in the following format:

```
Sun or Sunday
Mon or Monday
Tue or Tuesday
Wed or Wednesday
Thu or Thursday
Fri or Friday
Sat or Saturday
```

If you selected *In Range* as the *Comparison* type, specify the first day of the range in the left text box and the end day of the range in the right text box.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to a day of the week, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.6 Current Day of Month Condition

The *Current Day of Month* condition allows you to restrict access based on the day of the month the request is made. Fill in the following fields:

Comparison: Specify how the current day of the month is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: Day of Month:** Specifies that you want the values compared as a day of the month. Select one of the following operators:
  - **Equals:** Allows you to specify a day that the client must match.
  - **In Range:** Allows you to specify a range of days that the client’s request must fall within.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the second value for the comparison:

- If you select *Equals* for the *Comparison* type, you would normally select *Data Entry Field* for the *Value* field and specify a number from 1 to 31 in the text box.

- If you select *In Range* for the *Comparison* type, you would normally select *Data Entry Field* for the *Value* field and specify the first value of the range in the first text box and the second value of the range in the second text box. If you specify 1 in the first box and 15 in the second box, you can use this condition to restrict access between the first day of the month and the 15th day.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to a day of the month, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.7 Current Time of Day Condition

The *Current Time of Day* condition allows you to restrict access based on the time the request is made. Fill in the following fields:

Comparison: Specify how the current time of day is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: Time:** Specifies that you want the values compared as time. Select one of the following:
  - **Greater Than:** Requires that the current time is greater than the specified value.
  - **Greater Than or Equal to:** Requires that the current time is greater than or equal to the specified value.
  - **Less Than:** Requires that the current time is less than the specified value.
  - **Less Than or Equal to:** Requires that the current time is less than or equal to the specified value.
  - **In Range:** Requires that the current time must fall within the specified range, such as 08:00 and 17:00.

  If you specify this type of comparison, you must also specify a *Time Zone*. Select either the *Local* time zone or *GMT* (Greenwich Mean Time).
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the second value for the comparison. If you select *Data Entry Field* as the value type, hours and minutes are specified in the following format:

`hour:minute`

Hour is a number from 00 to 23, and minute is a number from 00 to 59.

Time can only be specified in a 24-hour clock format. For example, 8 am is 08:00 and 5:30 pm is 17:30.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to the time of day, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.8 HTTP Request Method Condition

The *HTTP Request Method* condition allows you to restrict accessed based on the request method in the current request.

HTTP Request Method: Select the request method from the list or select *Current* to specify the method in the current request.

Comparison: Specify how the HTTP Request Method is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the *HTTP Request Method* value must begin with the letters specified in the *Value* field.
  - **Ends with:** Indicates that the *HTTP Request Method* value must end with the letters specified in the *Value* field.
  - **Contains Substring:** Indicates that the *HTTP Request Method* value must contain the letters, in the same sequence, as specified in the *Value* field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode

Unix Lines

Value: Specify the value you want compared to the *HTTP Request Method* value. If you selected a method from the list for the *HTTP Request Method* value, select *HTTP Request Method* > *Current*. If you selected *Current* for the *HTTP Request Method* value, select a request method from the list.

Other value types are possible. Your policy requirements determine whether they are useful. If you have set up a Liberty User Profile or an LDAP attribute that corresponds to an HTTP Request Method, you can use this option and select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.9 LDAP Attribute Condition

The *LDAP Attribute* condition allows you to restrict access based on a value in an LDAP attribute defined for the inetOrgPerson class or any other LDAP attribute you have added. You can have the user’s attribute value retrieved from your LDAP directory and compared it to a value of the following type:

- Roles
- Date and time and its various elements
- URL and its various elements
- IP address
- Authentication contract
- Credential profile
- HTTP request method
- Liberty User Profile attribute
- Static value

To set up the matching for this condition, fill in the following fields:

LDAP Attribute: Specify the LDAP attribute you want to use in the comparison. Select from the listed LDAP attributes. To add an attribute that isn’t in the list, scroll to the bottom of the list, click *New LDAP Attribute*, then specify the name of the attribute.

Comparison: Specify how you want the values compared. All data types are available. Select one that matches the value type of your attribute.

Mode: Select the mode, if available, that matches the comparison type. For example, if you select to compare the values as strings, you can select either a *Case Sensitive* mode or a *Case Insensitive* mode.

Value: Specify the second value for the comparison. All data types are available. For example, you can select to compare the value of one LDAP attribute to the value of another LDAP attribute. Only you can determine if such a comparison is meaningful.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.10 LDAP OU Condition

The LDAP OU condition allows you to compare the DN of an OU against the DN that was used when the user authenticated. If the user's DN contains the OU, the condition matches.

LDAP OU: Select *[Current]*. This is the only option available if your Administration Console and Identity Server are installed on separate machines.

Comparison: Specify how you want the values compared. Select one of the following:

- **Contains:** Specifies that you want the condition to determine whether the user is contained by a specified organizational unit.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type.

- **Contains:** Select whether the user must be contained in the specified OU (*One Level)* or whether the user can be contained in the specified OU or a child container (*Subtree)*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the second value for the comparison. If your Administration Console and Identity Server are installed on separate machines, select *Data Entry Field* and enter the DN of the OU in the text field. If your Administration Console and Identity Server are installed on the same machine, you can select *LDAP OU*, then browse to the DN of the OU.

If you have defined a Liberty User Profile or an LDAP attribute for the OU you want to match, select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.11 Liberty User Profile Condition

The *Liberty User Profile* condition allows you to restrict access based on a value in a Liberty User Profile attribute. The Liberty attributes must be enabled before you can use them in policies (click *Identity Servers* > *Edit* > *Liberty* > *Web Server Provider*, then enable one or more of the following: *Custom Profile*, *Employee Profile*, *Personal Profile*).

These attributes can be mapped to LDAP attributes (click *Identity Servers* > *Edit* > *Liberty* > *LDAP Attribute Mapping*). When mapped, the actual value comes from your user store. If you are using multiple user stores with different LDAP schemas, mapping similar attributes to the same Liberty User Profile attribute allows you to create one policy with the Liberty User Profile attribute rather than multiple policies for each LDAP attribute.

The selected attribute is compared to a value of the following type:

- Roles
- Date and time and its various elements
- URL and its various elements
- IP address
- Authentication contract
- Credential profile
- HTTP request method
- LDAP attribute
- Static value

To set up the matching for this condition, fill in the following fields:

Liberty User Profile: Select the Liberty User Profile attribute. These attributes are organized into three main groups: Custom Profile, Corporate Employment Identity, and Entire Personal Identity. By default, the Common Last Name attribute for Liberty User Profile is mapped to the sn attribute for LDAP. To select this attribute for comparison, click *Entire Personal Identity* > *Entire Common Name* > *Common Analyzed Name* > *Common Last Name*.

Comparison: Select the comparison type that matches the data type of the selected attribute and the value.

Mode: Select the mode, if available, that matches the data type. For example, if you select to compare the values as strings, you can select either a *Case Sensitive* mode or a *Case Insensitive* mode.

Value: Select one of the values that is available from the current request or select *Data Entry Field* to enter a static value. The static value that you can enter is dependent upon the comparison type you selected.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.12 Roles for Current User Condition

If you have configured some Access Manager role policies (see Section 27.2, "Creating Roles," on page 394), you can use these roles as conditions to control access. Roles are not assigned to users until the users authenticate. All authenticated users are assigned the authenticated role. If you use a comparison type of starts with, ends with, or contains substring, carefully evaluate the potential results. For example, if you specify `ed` as the value for an ends with comparison, the condition matches roles such as contracted and assigned that you created, but it also matches the authenticated role.

Fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: String:** Specifies that you want the values compared as strings, and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the values must match, letter for letter.
  - **Starts with:** Indicates that the *Roles for Current User* value must begin with the letters specified in the *Value* field.
  - **Ends with:** Indicates that the *Roles for Current User* value must end with the letters specified in the *Value* field.
  - **Contains Substring:** Indicates that the *Roles for Current User* value must contain the letters, in the same sequence, as specified in the *Value* field.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: If you have created Identity Server roles policies, select *Roles*, then the role. The *authenticated* role is assigned to all users when they authenticate. If you have defined a Liberty User Profile or an LDAP attribute for roles, select this option, then select your attribute.

You can use the *Data Entry Field* option to enter the name of the role you want to test for.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.13 URL Condition

The *URL* condition allows you to restrict access based on the URL specified in the request. If you have users requesting a resource with a URL you don't want them to use, you can use this condition in an Access Gateway Authorization policy to deny them access to this URL, and use the Actions section to redirect the request to the URL you want them to use. In a J2EE Agent policy, you can only deny or allow; you cannot redirect.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the URL is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: URL: Equals:** Specifies that you want the values compared as URLs.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL: Equals:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: To enter a static value to compare to the URL in the current request, select *Data Entry Field* and specify the URL. This should be the complete URL, starting with the URL scheme (http:// or https://) and including the domain name. If the URL contains a path, you must include it. If you do not specify a scheme, http is used.

If you have selected *Regular Expression: Matches*, regular expression rules apply.

If you have selected *URL: Equals* for your comparison type, the wildcard characters (?) or (*) can be specified as the last element of the URL path to aid in matching basic URL patterns. These wildcard characters are interpreted as follows:

- ? matches all files at the specified directory level
- * matches all files and directories at and beyond the specified directory level

For example, if the request URL is `http://www.resourcehost.com/path/resource.gif`, the following entered URLs would match the request URL:

```
http://www.resourcehost.com/path/resource.gif
http://www.resourcehost.com/path/?
http://www.resourcehost.com/path/*
http://www.resourcehost.com/*
```

If you selected *URL:Equals* for the comparison type, you can add multiple values:

- Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
- Use the *Add* button to add values one at a time.
- Use the *URL Dredge* button to dredge for links to use as values. If you attempt to dredge an HTTPS site that is using a self-signed certificate, you need to import the trusted root of the site into the Trusted Roots store of the Access Gateway before performing the dredge.

All entered URLs are compared to the request URL until a match is found or the list is exhausted.

If you have defined a Liberty User Profile or an LDAP attribute for a URL, you can select these options for the value type, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.14 URL Scheme Condition

The *URL Scheme* condition allows you to restrict access based on the scheme specified in the URL of the request. For example in an Access Gateway Authorization policy, if the request contains HTTP as the scheme in the URL and you require them to use HTTPS, you can use this condition to deny access and redirect them to another URL. In a J2EE Agent policy, you can only deny or allow; you cannot redirect.

This condition allows you to compare A to B or B to A. You need to decide whether you want to compare a static value to the current value in the HTTP request, or whether you want to compare the current value in the HTTP request to a specified value. The comparison type you use depends upon the value you want to specify. If you want more flexibility in specifying the value, you should select to compare the current value in the HTTP request with a specified value.

To set up matching for this condition, fill in the following fields:

URL Scheme: Specify the scheme you want compared. You can select *Current* for the current value in the HTTP request, or specify a static value of *http* or *https*.

Comparison: Select one of the following types:

- **Comparison: URL Scheme:** Specifies that you want the values compared as scheme strings and how you want the values compared. Select one of the following:
  - **Equals:** Indicates that the URL scheme must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL scheme must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL scheme must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the URL scheme must contain the letters specified in the value.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: String:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All

Multi-Line
Unicode
Unix Lines

Value: Specify the value you want to compare with the *URL Scheme* value. If you select a static value for the *URL Scheme* value, select *URL Scheme* and *Current*. If you select *Current* for the *URL Scheme* value, select one of the following value types:

- **Data Entry Field:** Allows you to specify the scheme value you want to use in the comparison. The scheme cannot be specified with a trailing colon (:) character and must be specified in lower case (*http* or *https*). Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line. Use the *Add* button to add values one at a time.

  All entered URL schemes are compared to the requested URL scheme until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL scheme, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL scheme, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.15 URL Host Condition

The *URL Host* condition allows you to restrict access based on the hostname specified in the URL of the request. For example, you can use this condition to create rules that allow access if the URL contains one hostname, but deny access if the URL contains another hostname. The URL Host condition compares the hostname in the URL of the current request to the URL hostname specified in the *Value* field.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the URL Host is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: URL Host: Equals:** Specifies that you want the values compared as hostnames.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Select one of the following value types, then specify a value:

- **Data Entry Field:** To specify a static value to compare to the URL host in the current request, select this value type and specify the DNS name of the host.

  For example, if the request URL is http://www.resourcehost.com/path/resource.gif, the following hostname would match the resource URL:
  `www.resourcehost.com`

  If you selected *URL Host:Equals* for the comparison type, you can add multiple values:
  - Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
  - Use the *Add* button to add values one at a time.

  All listed hostnames are compared to the requested URL until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL host, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL host, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.16 URL Path Condition

The *URL Path* condition allows you to restrict access based on the path specified in the URL of the request. This condition compares the path of the URL in the current request to the path specified in the *Value* field.

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL Path:** Specifies that you want the values compared as paths and how you want the string values compared. Select one of the following:
  - **Equals:** Indicates that the URL path must contain the same letters, in the same order as specified in the value.
  - **Starts with:** Indicates that the URL path must begin with the letters specified in the value.
  - **Ends with:** Indicates that the URL path must end with the letters specified in the value.
  - **Contains Substring:** Indicates that the URL path must contain the letters specified in the value.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL Path:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.

- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To enter a static value to compare to the URL path in the current request, select this value type and specify the path. Start the path with a forward slash.

  If you have selected *Regular Expression: Matches* for your comparison type, regular expression rules apply. If you have selected *URL Path* for your comparison type, the path can end with a filename or a wildcard. An asterisk (*) matches all files and directories at and beyond the specified directory level. A question mark (?) matches all files at the specified directory level. For example:

Path	Match Description
`/path1/path2/`	Requires an exact match of the URL path. It matches if the URL does not contain anything other than `path2`.
`/path1/file.ext`	Requires an exact match of the URL path, including the extension on the filename.
`/path1/path2/?`	Matches everything that immediately follows `path2`. It does not match anything if the path contains another directory, such as `/path1/path2/path3/file3.ext`.
`/path1/path2/*`	Matches everything that follows `path2`, including a filename or another directory, such as `/path1/path2/path3/file3.ext`.

  If you selected *URL Path* for the comparison type, you can add multiple values:

  - Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
  - Use the *Add* button to add values one at a time.

  All entered URL paths are compared to the request URL path until a match is found or the list is exhausted.

- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or URL path, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or URL path, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.17 URL File Name Condition

The *URL File Name* condition allows you to restrict access based on the filename specified in the URL. It compares the filename in the URL of the current request to the filename specified in the *Value* field.

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL File:** Specifies that you want the values compared as filenames and how you want the names compared. Select one of the following:
    - **Equals:** Indicates that the filenames must contain the same letters, in the same order as specified in the value.
    - **Starts with:** Indicates that the filenames must begin with the letters specified in the value.
    - **Ends with:** Indicates that the filenames must end with the letters specified in the value.
    - **Contains Substring:** Indicates that the filenames must contain the letters specified in the value.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL File:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To specify a static value to compare to the filename in the current request, select this value type and specify the filename.

  The value you specify is compared to what follows the last slash in the URL. If you selected *Regular Expression: Matches* for your comparison type, regular expression rules apply. If you selected *URL File* for your comparison type, enter a value that matches your string comparison type. Do not use wildcards in your value.

  If you selected *URL File* for the comparison type, you can add multiple values:
    - Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
    - Use the *Add* button to add values one at a time.

All listed filenames are compared to the requested URL filename until a match is found or the list is exhausted.

- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or filename, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or filename, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.18 URL File Extension Condition

The *URL File Extension* condition allows you to restrict access based on the file extension specified in the URL of the request. It compares the file extension in the URL of the current request to the extension specified in the *Value* field

To set up matching for this condition, fill in the following fields:

Comparison: Select one of the following types:

- **Comparison: URL File:** Specifies that you want the values compared as file extensions and how you want the file extensions compared. Select one of the following:
    - **Equals:** Indicates that the file extensions must contain the same letters, in the same order as specified in the value.
    - **Starts with:** Indicates that the file extensions must begin with the letters specified in the value.
    - **Ends with:** Indicates that the file extensions must end with the letters specified in the value.
    - **Contains Substring:** Indicates that the file extensions must contain the letters specified in the value.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions.

Mode: Select the mode appropriate for the comparison type:

- **Comparison: URL File Extension:** Specify whether case is important by selecting *Case Sensitive* or *Case Insensitive*.
- **Comparison: Regular Expression: Matches:** Select one or more of the following:

  Canonical Equivalence
  Case Insensitive
  Comments
  Dot All
  Multi-Line
  Unicode
  Unix Lines

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To specify a static value to compare to the file extension in the current request, select this value type and specify the file extension. You can specify the extension or the period and the extension. For example:

  ```
  .ext
  ext
  ```

  This condition does not support wildcards. If you selected *URL File Extension* for the comparison type, you can add multiple values:

  - Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
  - Use the *Add* button to add values one at a time.

  All entered URL file extensions are compared to the requested URL file extension until a match is found or the list is exhausted.
- **LDAP Attribute:** If you have defined an LDAP attribute containing a URL or file extension, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute containing a URL or file extension, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.5.19 X-Forward-For IP Condition

For added security, you can add the IP address of the reverse proxy as a condition to check before granting access. One way to implement this is to create a rule that requires the X-Forwarded-For IP address in the HTTP header to match the configured IP address of the reverse proxy that is using the policy. The X-Forwarded-For IP condition matches the first IP address in the X-Forwarded-For header with the IP address specified in the *Value* field.

To set up matching for this condition, fill in the following fields:

Comparison: Specify how the X-Forwarded-For IP address is compared to the data in the *Value* field. Select one of the following types:

- **Comparison: IP:** Specifies that you want the values compared as IP addresses. Select one of the following:
  - **Equals:** Allows you to specify an IP address that the X-Forwarded-For IP address must match. You can specify more than one.
  - **In Range:** Allows you to specify a range of IP addresses that the X-Forwarded-For IP address must fall within. You can specify more than one range.
  - **In Subnet:** Allows you to specify the subnet that the X-Forwarded-For IP address must belong to. You can specify more than one subnet.
- **Comparison: Regular Expression: Matches:** Specifies that you want the values compared as regular expressions. If you select this option, you must also specify a *Mode*. Select one or more of the following:

Canonical Equivalence
Case Insensitive
Comments
Dot All
Multi-Line
Unicode
Unix Lines

Value: Specify the value type and value for the comparison. Select one of the following:

- **Data Entry Field:** To specify a static value, select *Data Entry Field* and provide a value appropriate for your comparison type. For example:

Comparison Type	Value

Equals	`10.10.10.10`
	`10.10.10.11`

In Range	`10.10.10.10 - 10.10.10.100`
	`10.10.20.10 - 10.10.20.100`

In Subnet	`10.10.10.12 / 22`
	`10.10.20.30 / 22`

  If you selected *IP* for the comparison type, you can add multiple values:

  - Use the *Edit* button to access a text box where you can enter multiple values, each on a separate line.
  - Use the *Add* button to add values one at a time.

  All listed values are compared to the IP address in the header until a match is found or the list is exhausted.

- **Client IP:** If you want the first IP address in the X-Forwarded-For header compared to the IP address of the client making the request, select this option.
- **LDAP Attribute:** If you have defined an LDAP attribute for an IP address, you can select this option, then select your attribute.
- **Liberty User Profile:** If you have defined a Liberty User Profile attribute for an IP address, you can select this option, then select your attribute.

Result on Condition Error: Specify what the condition returns when the comparison of the two values returns an error rather than the results of the comparison. Select either *False* or *True*. If you do not want the action applied when an error occurs, select *False*. If you want the action applied when an error occurs, select *True*.

28.6 Sample Policies

The following samples are provided so you can comprehend the variety of ways you can use the conditions to control access. The biggest difference between a J2EE Agent Authorization policy and an Access Gateway Authorization policy is that an Access Gateway Authorization policy allows a redirect action. J2EE Agent Authorization policies allow only a deny or permit action.

- Section 28.6.1, "Sample Policy Based on Organizational Rules," on page 449

- Section 28.6.2, “Sample Workflow Policy,” on page 452
- Section 28.6.3, “Sample Date and Time Policy,” on page 455
- Section 28.6.4, “Controlling Access Based on IP Addresses and Roles,” on page 458

28.6.1 Sample Policy Based on Organizational Rules

The following sections describe a scenario with an organizational division, then describe two types of policies that enforce the requirements of the scenario. The policy can be created as an Access Gateway Authorization policy and as a J2EE Agent Authorization policy:

- “Company Scenario” on page 449
- “LDAP Context Policies” on page 449
- “Role Policies with Authorization Policies” on page 450

Company Scenario

Suppose that the company LDAP directory has the following organization.

ou=sales,o=acme
ou=dev,o=acme
ou=hr,o=acme

Suppose that this company has the following configuration and requirements:

- Under each branch of the tree, the system administrator has created the users who work in these departments.
- Each department has its own Web resources, and other departments must be denied access to these resources.

With this type of configuration, you can use the LDAP context condition to create authorization policies or you can create role policies that are used in conjunction with authorization policies.

LDAP Context Policies

With such an organization, you can create a policy that either allows or denies access based on the LDAP context of the user’s DN. You can use the LDAP context of the user DN to separate the users into their departments and then grant access based on the context match. You need to create protected resources for the Web resources of the department, create a policy for each protected resource, and assign a policy to the protected resources.

The following procedure explains how to configure such a policy for the sales department.

1. Click *Policies* > *New*, specify a name for the policy, select *Access Gateway: Authorization* as the type, then click *OK*.
2. For *Condition Group 1*, click *New*, then select *Credential Profile*.
3. Fill in the following fields:

   LDAP Credentials: Select *LDAP User DN*.

   If/If Not: Select *If Not*.

   Comparison: Select *Contains Substring*.

Mode: Select *Case Insensitive.*

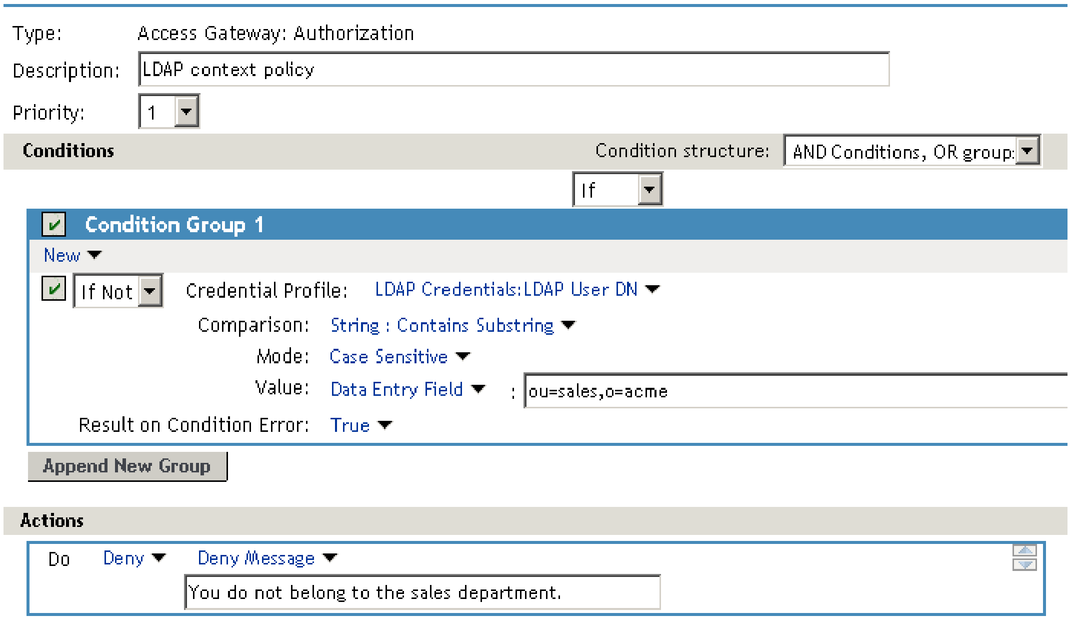
Value: Select *Data Entry Field*. In the text box, type the following value:

`ou=sales,o=acme`

Result on Condition Error: Select *True*.

4 In the *Actions* section, select *Deny*.

Your policy should look similar to the following:



Type: Access Gateway: Authorization

Description: LDAP context policy

Priority: 1

Conditions — Condition structure: AND Conditions, OR group

If

☑ Condition Group 1

New

☑ If Not — Credential Profile: LDAP Credentials:LDAP User DN

Comparison: String : Contains Substring

Mode: Case Sensitive

Value: Data Entry Field : ou=sales,o=acme

Result on Condition Error: True

Append New Group

Actions

Do Deny — Deny Message: You do not belong to the sales department.

This sets up the condition so that the following occurs:

- When the user does not belong to the sales department, the user is denied access.
- When the user belongs to the sales department, the user is granted access.
- When an error occurs evaluating the conditions in the rule, the user is denied access.

5 Assign the policy to the protected Web resources of the sales department (see Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210).

6 Repeat these steps for the other two departments, changing the *Value* field to match the appropriate department.

Role Policies with Authorization Policies

Because of the company’s organization, you need to create three role policies, one for the sales users, one for the development users, and one for the human resource users. You can then use these roles as conditions in authorization policies to allow and deny access. The first time you use roles in an authorization policy, there is extra setup because you must create the role policies. However after the role policies are created, you can use them in multiple authorization policies.

The following instructions explain how to use the Sales role to create a policy that controls access to a protected resource. For instructions on how to create the Sales role, see Section 27.3.2, “Creating a Role Using the Location of the User Objects,” on page 405.

You need to decide on the type of Authorization policy you want to create. For example, you can create a Deny policy that denies access to everyone who does not match the condition (in this case, the Sales role). Or you can create a two-rule policy that allows access to everyone that matches the condition. The first rule grants access to everyone who has the Sales role, and the second rule denies access to everyone who did not match the conditions of the first rule. (Other methods are also possible.) Because the proposed Deny policy is very similar to the LDAP Context Policies example, the following procedures explain how to create the two-rule policy.

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *Access Gateway: Authorization* as the type, then click *OK*.
3. (Optional) Provide a description for the rule.
4. In *Condition Group 1*, click *New*, and select *Roles for Current User*.
5. Fill in the following fields:

   If/If Not: Select *If.*

   Comparison: Select *String: Equals*.

   Mode: Select *Case Insensitive*

   Value: Select *Roles,* then select *Sales.*

   Result on Condition Error: Select *False*.
6. Under *Actions*, select *Permit*, then click OK.

   These steps create the Permit rule and set up the condition so that the following occurs:

   - When the user does not match the condition because the user does not belong to the Sales role, the policy engine moves to the next rule in the policy.
   - When the user does match the condition because the user belongs to the Sales role, the user is granted access.
   - If an error occurs when evaluating the condition of the policy, the user does not match the condition and the policy engine moves to the next rule in the policy.
7. In the *Rule List*, click *New*.

   This second rule is for denying access to everyone who does not match the condition in Rule 1. Processing of the policy stops when a user matches a rule; therefore all users who match Rule 1 are granted access and the policy engine does not evaluate the second rule.
8. Set the *Priority* to be 2 or greater.

   You want the Permit rule to be processed first, so it should have a priority of 1. The Deny rule needs to be processed last, so it needs a lower priority than the Permit rule.
9. Leave the *Condition Group 1* empty.

   The *Conditions* section is left empty so that everyone who does not match the conditions of the Permit rule is denied access to the resource.
10. In the *Actions* section, select *Deny* and either accept the default action or select one of the other actions.
11. Click *OK* twice.
12. Click *Apply Changes* on the Policies page.
13. Assign the policy to the protected Web resources of the sales department (see Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210).

28.6.2 Sample Workflow Policy

This sample policy can only be created as an Access Gateway Authorization policy because it relies on a redirect action.

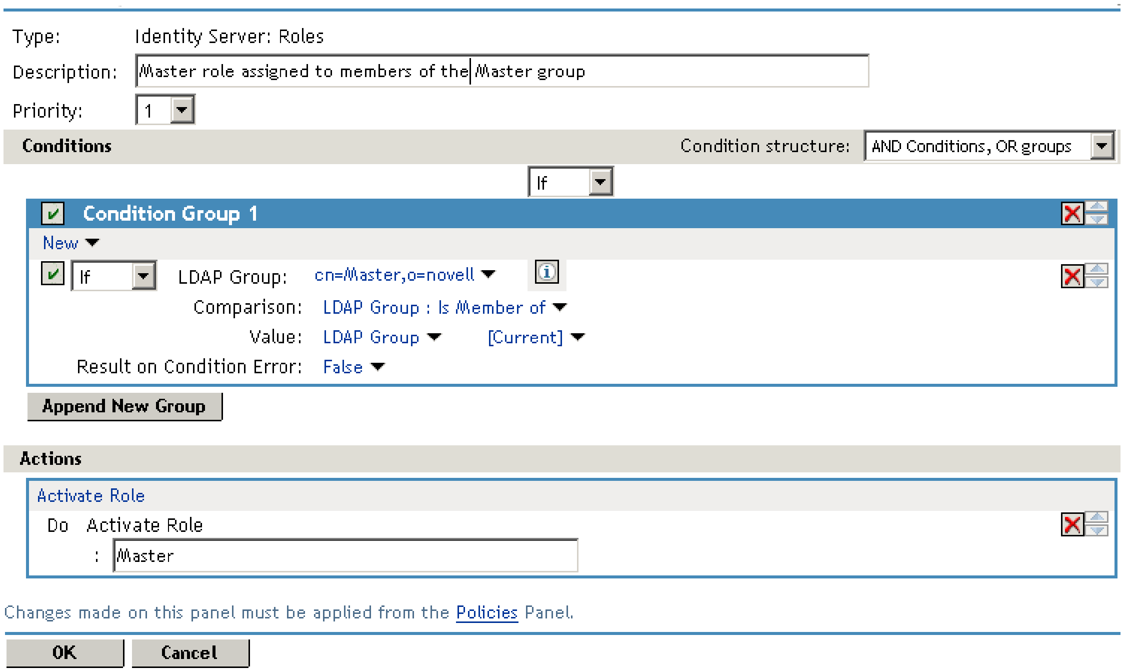
One of the common workflow problems that an Authorization policy can solve is what to do with users who are denied access to resource. Most of the time they have a legitimate reason for trying to access the resource and need contact information to request access to the resource. You can add this contact information to a Web page and redirect the users to this page when the policy denies the user access.

To create such a workflow, you need to create an HTML page with the necessary information for making the request for access. It can be as simple as a contact name or it can be an actual form that the user submits to the organization that controls access to the resource.

You then need to create an Authorization policy that redirects the denied users to this page. The following sample policy uses a role for the access condition, but the same workflow can be created using any of the other conditions available for an Authorization policy. For this example, let's assume that the user is granted a Master role if the user is a member of the Master group. The organization that controls access to the resource is the owner of the Master group and can add and delete members from the group. When the owner of the Master group receives a request for access to the resource, the owner can evaluate the user, and if the user meets their standards, the owner adds the user to the Master group.

You can use the Master group to create an Access Manager Role policy. This policy for the Master role should look similar to the following:

***Figure 28-8** Role Policy with an LDAP Group Condition*



Type: Identity Server: Roles

Description: Master role assigned to members of the Master group

Priority: 1

Condition structure: AND Conditions, OR groups

If LDAP Group: cn=Master,o=novell

Comparison: LDAP Group : Is Member of

Value: LDAP Group [Current]

Result on Condition Error: False

Do Activate Role

: Master

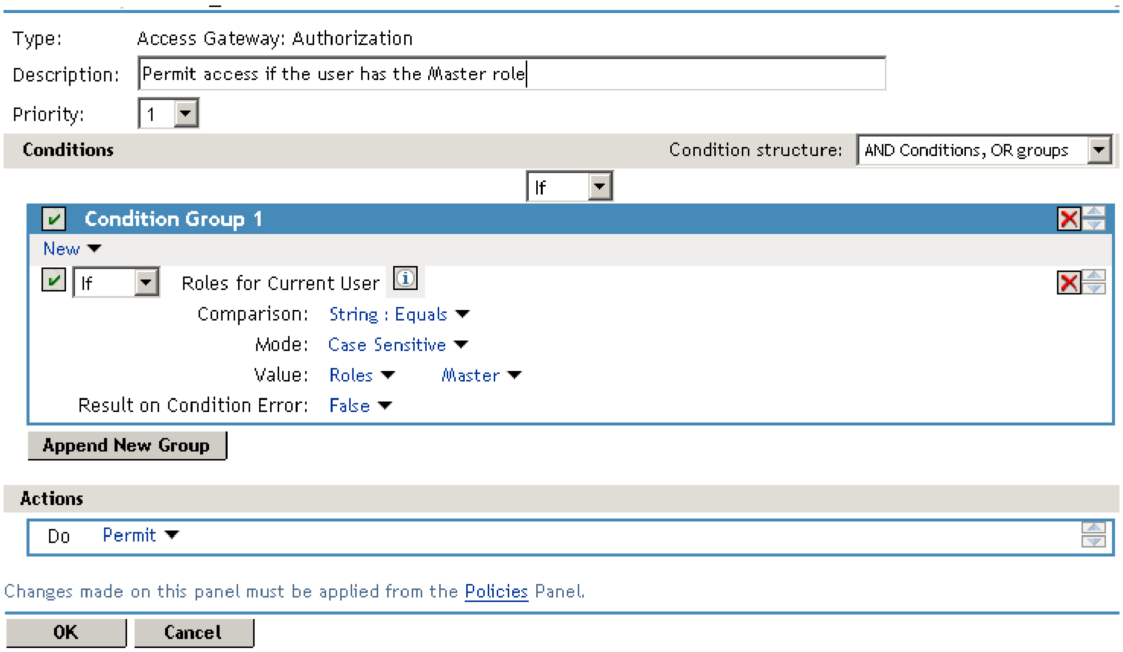
Changes made on this panel must be applied from the Policies Panel.

This rule grants the user the Master role if the user belongs to the cn=Master,o=novell LDAP group. If the user doesn't belong to this group or if an error occurs trying to get the data, the user is not assigned the role. This occurs because both the condition and the *Result on Condition Error* evaluate to False, which prevents the Action from being applied.

After creating the Role policy, apply the changes and enable the Role for the Identity Server.

You can then use this role to create an Authorization policy that contains two rules. The first rule grants access to the users who have the Master role (and are therefore members of the Master group). This rule should look similar to the following:

Figure 28-9 *A Permit Rule with a Role Condition*



Type: Access Gateway: Authorization

Description: Permit access if the user has the Master role

Priority: 1

Conditions — Condition structure: AND Conditions, OR groups

If

Condition Group 1

If Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles Master

Result on Condition Error: False

Actions

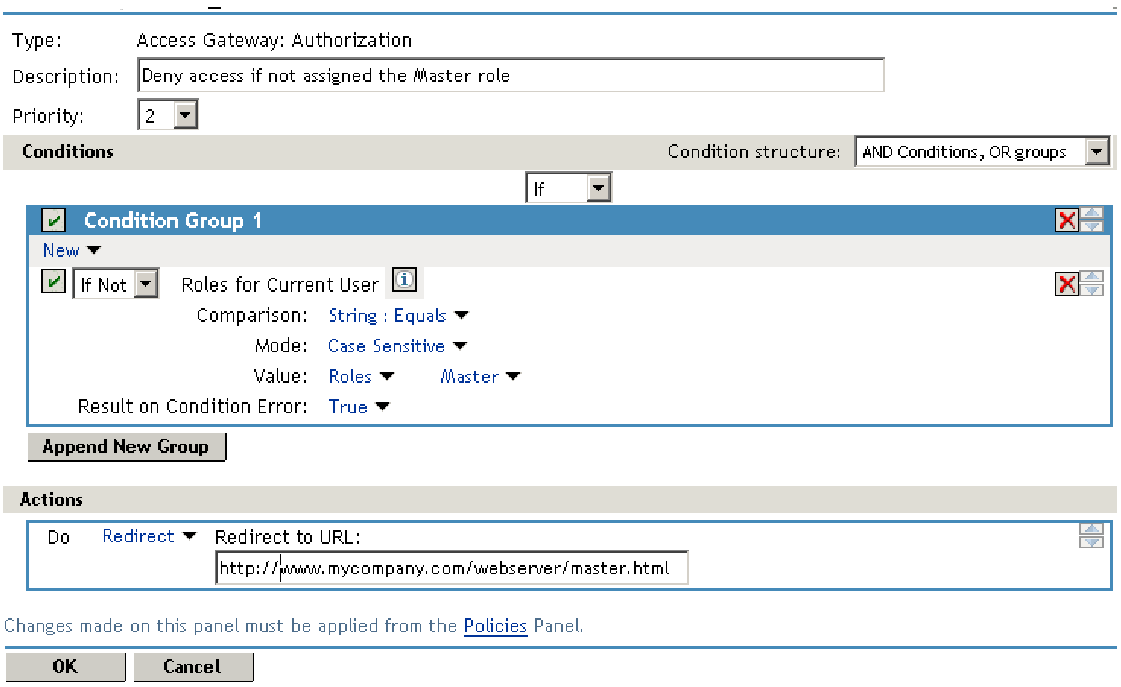
Do Permit

Changes made on this panel must be applied from the Policies Panel.

This rule permits users who are assigned the Master role to have access to the resource. If the user does not match the condition or if an error occurs accessing the user's role information, the user is sent to the next rule because both the condition and the *Result on Condition Error* evaluate to False.

The second rule in the policy should deny access to those who are not assigned the Master role and should redirect them to the page where they can request access. You can do this with a rule that checks to see if they are assigned the Master role. In this type of rule, the condition needs to be an *If Not* condition.

Figure 28-10 *A Deny Rule with a Redirect URL*



Type: Access Gateway: Authorization

Description: Deny access if not assigned the Master role

Priority: 2

Conditions Condition structure: AND Conditions, OR groups

If

Condition Group 1

New

If Not Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles Master

Result on Condition Error: True

Actions

Do Redirect Redirect to URL: http://www.mycompany.com/webserver/master.html

Changes made on this panel must be applied from the Policies Panel.

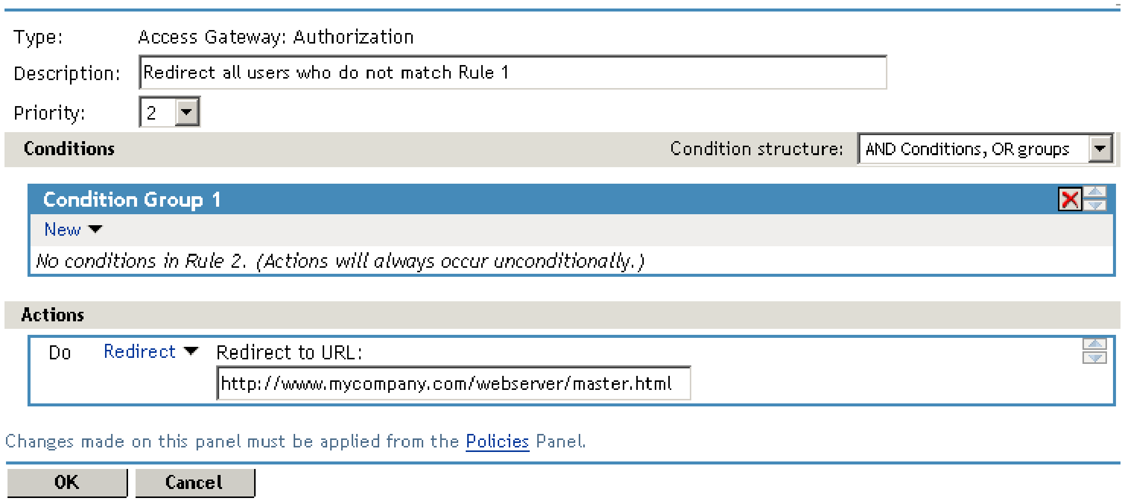
With an *If Not* condition, the condition evaluates to True when the user does not match the condition. With such a rule, you want the *Result on Condition Error* to also evaluate to True. If there is an error obtaining role information for the user, you don't want the rule to assume that the user had the Master role. You want the rule to assume that the user had no roles, or in other words, you want the error condition to evaluate to True.

Because the condition evaluated to True, the Action is applied to the user. The value specified in the *Redirect to URL* text box should specify the page that contains the information on how to request access.

This redirect rule could be the only rule in the Authorization policy, because the users who are assigned to the Master role do not match the rule and are thus allowed access. Having the first rule that grants access because they have the Master role just makes the logic of the policy clearer.

If you create the first rule that grants users with the Master role access, you can use a general Deny rule for the second rule. It should look similar to the following.

Figure 28-11 *A General Deny Rule*



Type: Access Gateway: Authorization

Description: Redirect all users who do not match Rule 1

Priority: 2

Condition structure: AND Conditions, OR groups

No conditions in Rule 2. (Actions will always occur unconditionally.)

Do Redirect Redirect to URL: http://www.mycompany.com/webserver/master.html

Changes made on this panel must be applied from the Policies Panel.

A general Deny rule has no conditions, so it matches everyone that does not match the first rule in the policy. You can add more rules to this policy to tighten security so that not all users are redirected to the site that contains the information on how to request access. For this type of policy, the last rule would be a general Deny rule with no conditions and without a redirect. The rules between Rule 1, which granted access to people assigned to the Master role, and the last rule, which denies everyone, should be rules that identify the types of users who have legitimate reasons for requesting access, and these rules should contain the redirect action.

After you have saved the Authorization policy, you need to assign it to the protected resource or resources that require the Master role, then update the Access Gateway.

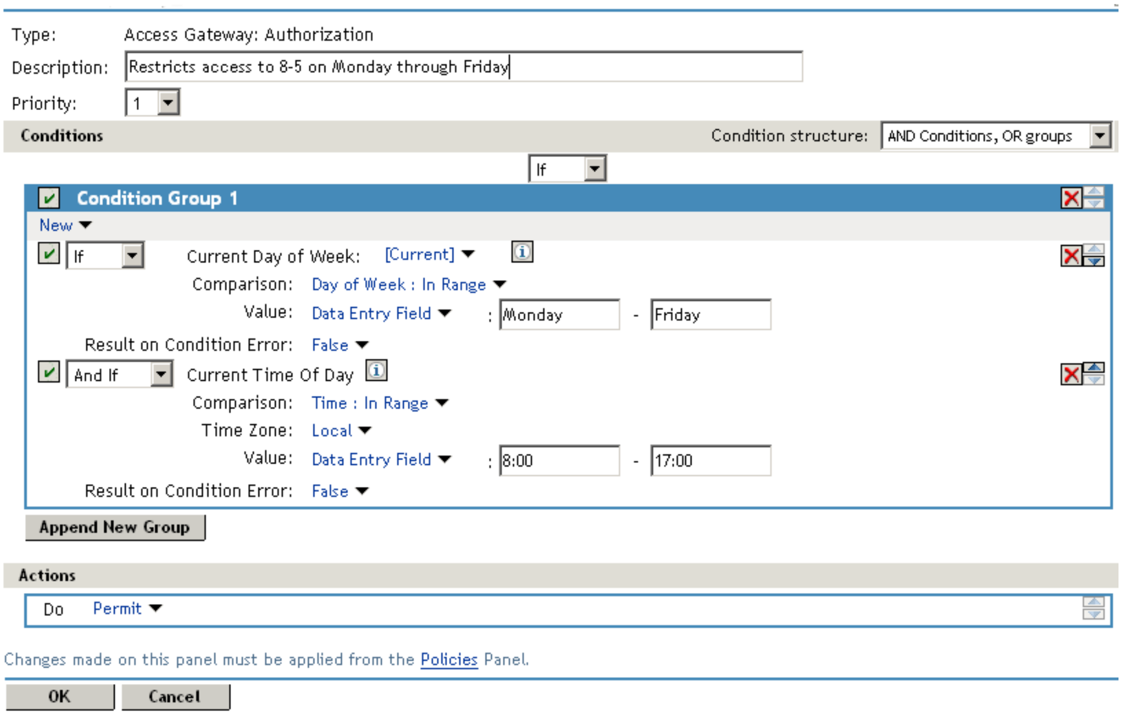
28.6.3 Sample Date and Time Policy

If you have an application or data that is highly sensitive, you might want to limit access so that it is only available during core work hours, for example Monday through Friday from 8:00 a.m. to 5:00 p.m. This type of policy can be created as either an Access Gateway Authorization policy or as a J2EE Agent Authorization policy.

The policy requires the use of the Current Day of Week and the Current Time of Day conditions.

The first rule in the policy has its priority set to 1, adds both conditions to Condition Group 1, uses a comparison type of In Range, and selects Data Entry Field for the value. Such a rule looks similar to the following:

Figure 28-12 *Day and Time Policy Rule*



Type: Access Gateway: Authorization

Description: Restricts access to 8-5 on Monday through Friday

Priority: 1

Conditions Condition structure: AND Conditions, OR groups

If

Condition Group 1

New

☑ If Current Day of Week: [Current]

Comparison: Day of Week : In Range

Value: Data Entry Field : Monday - Friday

Result on Condition Error: False

☑ And If Current Time Of Day

Comparison: Time : In Range

Time Zone: Local

Value: Data Entry Field : 8:00 - 17:00

Result on Condition Error: False

Append New Group

Actions

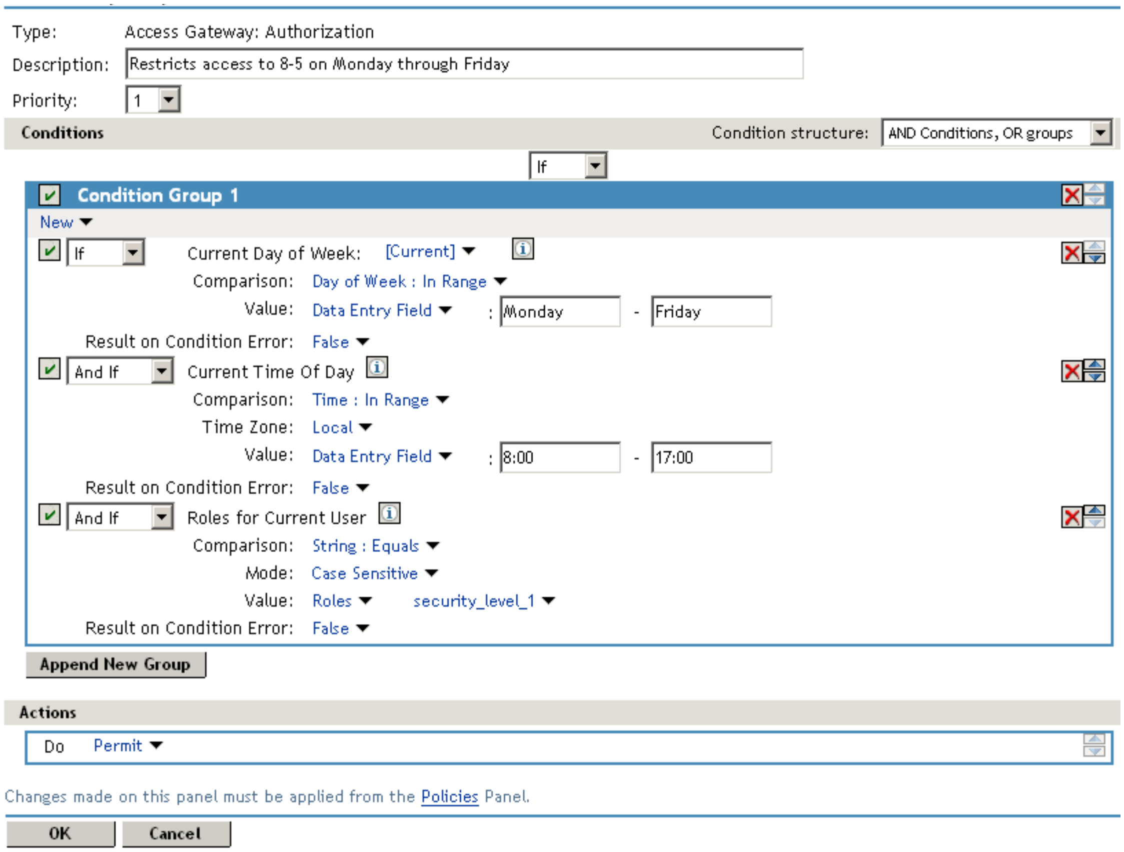
Do Permit

Changes made on this panel must be applied from the Policies Panel.

OK Cancel

You might need to add user conditions to this rule, if you don't want all authenticated users to have access to the resource. Figure 28-13 shows a condition added to the rule that requires users to have the security_level_1 role.

Figure 28-13 *Day and Time Rule with a Role Condition*



Type: Access Gateway: Authorization

Description: Restricts access to 8-5 on Monday through Friday

Priority: 1

Conditions Condition structure: AND Conditions, OR groups

If

Condition Group 1

New

If Current Day of Week: [Current]

Comparison: Day of Week : In Range

Value: Data Entry Field : Monday - Friday

Result on Condition Error: False

And If Current Time Of Day

Comparison: Time : In Range

Time Zone: Local

Value: Data Entry Field : 8:00 - 17:00

Result on Condition Error: False

And If Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles security_level_1

Result on Condition Error: False

Actions

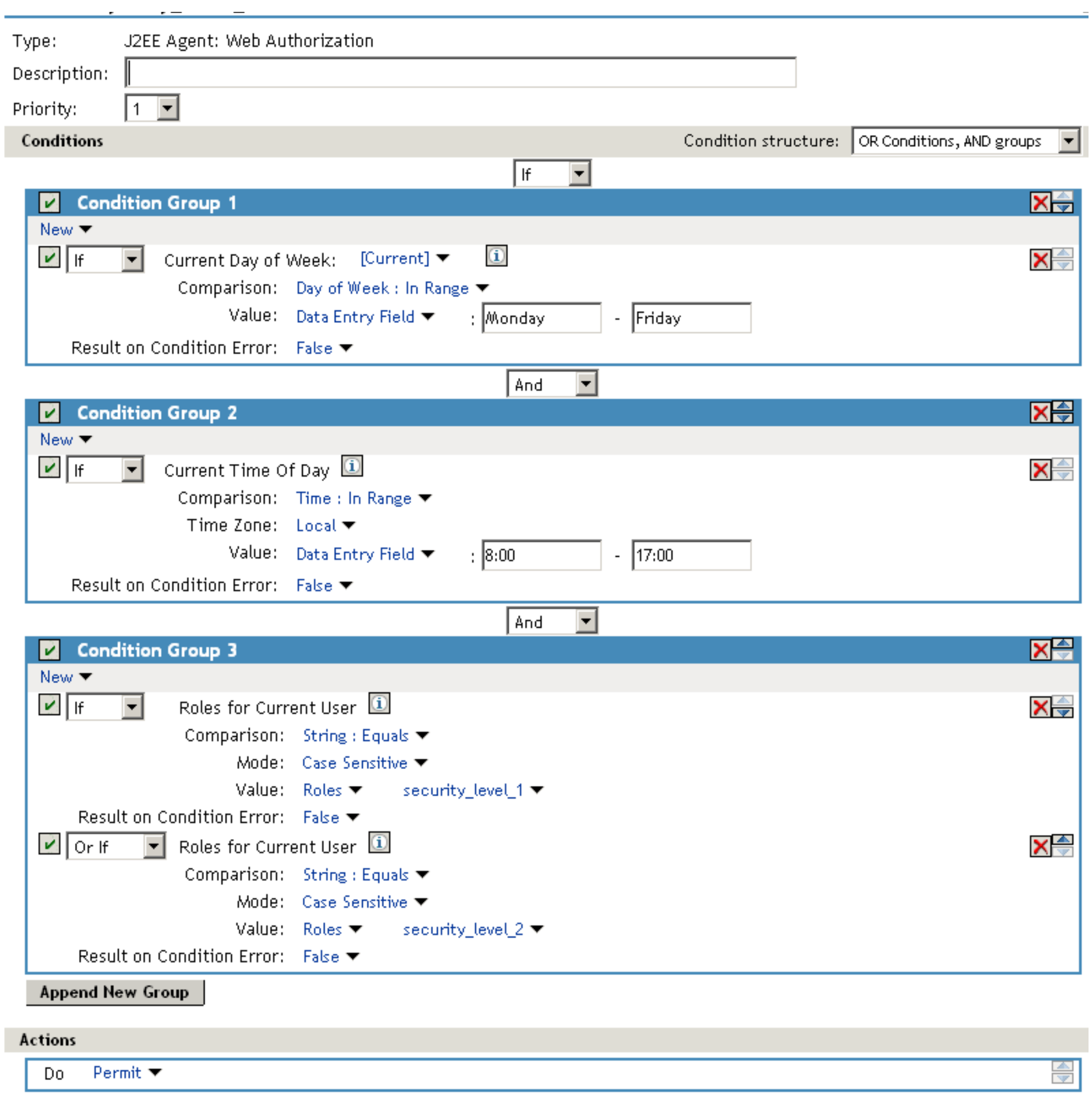
Do Permit

Changes made on this panel must be applied from the Policies Panel.

You can add multiple condition groups, each with the Current Day of Week and Current Time of Day conditions, and each with a different user condition that must be met to permit access. The rule is configured so that a user must match all the conditions in a Condition Group. The Condition Groups are ORed, so the user needs to match the conditions in only one Condition Group for a match.

The rule can be configured to OR conditions and AND groups. In this type of rule, the user must match at least one condition in each Condition Group. With this configuration, the rule looks similar to the following.

Figure 28-14 *A Rule with Multiple ANDed Condition Groups*



Type: J2EE Agent: Web Authorization

Description:

Priority: 1

Conditions

Condition structure: OR Conditions, AND groups

If

☑ Condition Group 1

☑ If Current Day of Week: [Current]
Comparison: Day of Week : In Range
Value: Data Entry Field : Monday - Friday
Result on Condition Error: False

And

☑ Condition Group 2

☑ If Current Time Of Day
Comparison: Time : In Range
Time Zone: Local
Value: Data Entry Field : 8:00 - 17:00
Result on Condition Error: False

And

☑ Condition Group 3

☑ If Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles security_level_1
Result on Condition Error: False

☑ Or If Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles security_level_2
Result on Condition Error: False

Actions

Do Permit

All users must match the conditions in Condition Group 1 and 2, but only one of the conditions in Condition Group 3.

If the user does not match conditions in Rule 1, the next rule in the policy is evaluated. You need to add a general deny rule set to a lower priority. A general deny rule has no conditions, and the action is set to deny. Thus all users who do not match Rule 1 match this rule and are denied access.

28.6.4 Controlling Access Based on IP Addresses and Roles

In this scenario, the company has multiple offices. Headquarters are in New York, and you want users at this office to authenticate using a username/password contract before accessing the corporate site. The users at branch offices need to use a smart card to authenticate before accessing the corporate site.

To create such a configuration, you need to create Web pages, an Access Gateway Authorization policy, protected resources, and a smart card contract.

- "Web Pages" on page 459
- "Protected Resources" on page 459
- "Characteristics of Rule 1" on page 459
- "Characteristics of Rule 2" on page 460
- "Characteristics of Rule 3" on page 461
- "Characteristics of Rule 4" on page 462
- "Other Considerations" on page 463

Web Pages

You need three Web pages:

- **Initial Access URL:** The content of this page isn't important. No one should ever see the page, but it should exist and should probably contain a warning that if you see this page, you should call the help desk.
- **Outside Headquarters Redirect URL:** This page needs to contain a redirect to the corporate site.
- **Corporate Site:** This should be the main page of your site, with links to the resources your users need.

Protected Resources

You need three protected resources with the following policy or contracts:

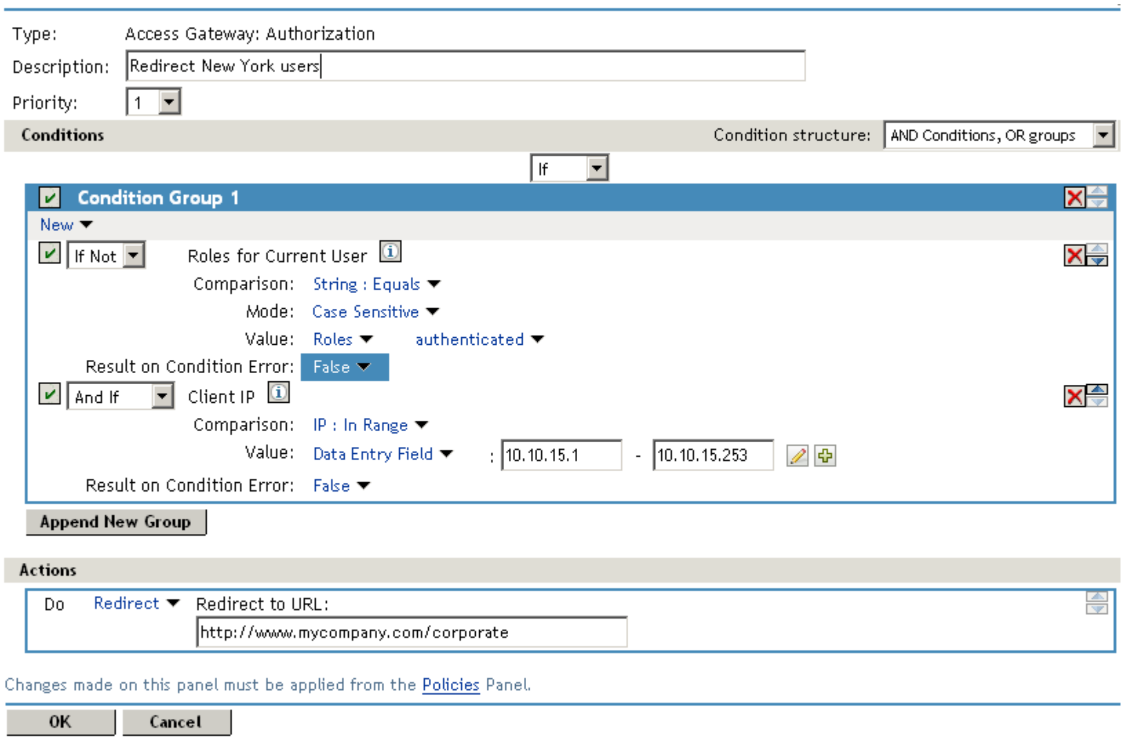
- **Initial Access URL:** With the Redirect policy, but no contract.
- **Outside Headquarters Redirect URL:** A smart card contract.
- **Corporate Site:** A username/password contract. The username/password contract needs to be configured so that it is satisfied by a contract of equal or higher value. If you set the username/password contract to an Authentication level of 1 and the smart card contract to an Authentication level of 2, the smart card contract is at a higher level and therefore satisfies the requirements of the username/password contract.

Characteristics of Rule 1

The first Web page needs to be a URL that all users specify to access the corporate site. This page needs to be protected by an Authorization policy with at least four rules. You need to configure Rule 1 to have the following conditions, actions, and characteristics:

Characteristics	User Experience
Condition Group 1 sets the following: - Uses the Roles for Current User condition to match users who do not have the authenticated role. - Uses the Client IP condition to match the users who have an IP address from the New York office. The Action is set to Redirect with a URL to the corporate site that is protected by a username/password contract. The priority of the Rule is set to 1. The rule description indicates that this rule redirects the New York users.	A New York user, who is unauthenticated, sees the redirect, is prompted for a username and password, then is granted access to the corporate site.

Figure 28-15 *Rule 1 with the Conditions for the New York Office*



Type: Access Gateway: Authorization
Description: Redirect New York users
Priority: 1

Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

☑ If Not Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles authenticated
Result on Condition Error: False

☑ And If Client IP
Comparison: IP : In Range
Value: Data Entry Field : 10.10.15.1 - 10.10.15.253
Result on Condition Error: False

Do Redirect Redirect to URL:
http://www.mycompany.com/corporate

Changes made on this panel must be applied from the Policies Panel.

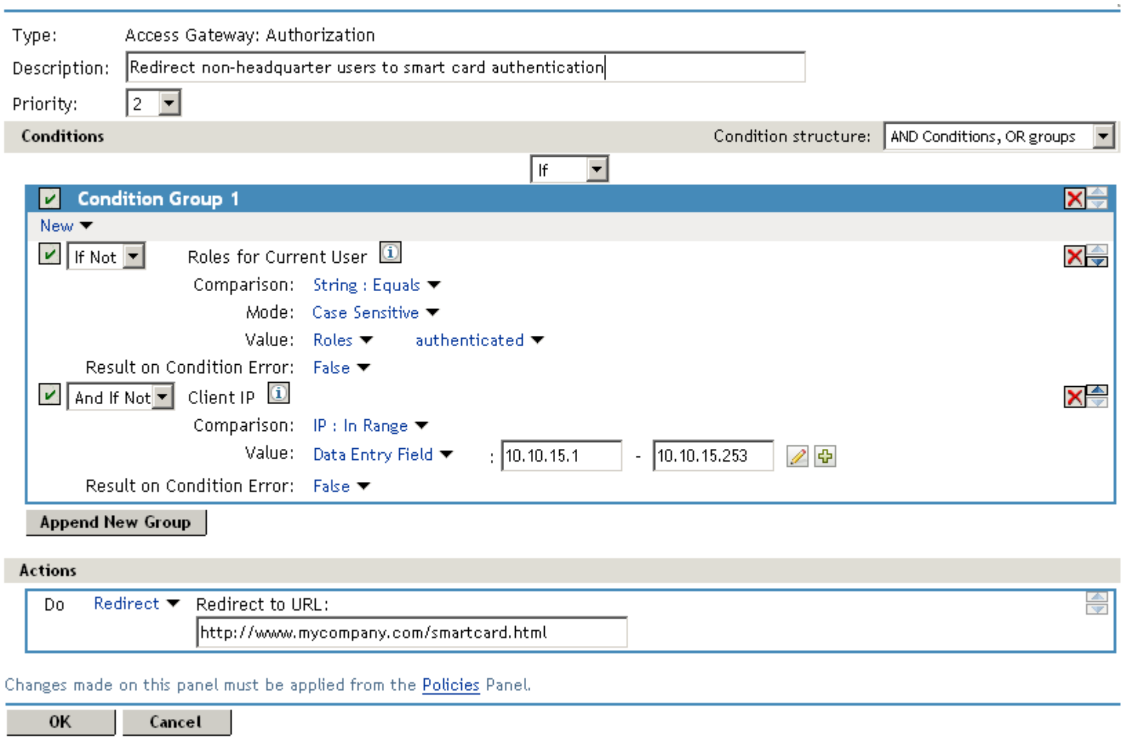
In Rule 1, the Roles for Current User condition is set to be an If Not condition so that users who are not authenticated match this condition. The second condition, Client IP, is an And If condition, because the IP address of the user needs to match a value in the Data Entry Field.

Characteristics of Rule 2

You need to configure Rule 2 to have the following conditions, actions, and other characteristics:

Characteristics	User Experience
Condition Group 1 sets the following: ◆ Uses the Roles for Current User condition to match users who do not have the authenticated role. ◆ Uses the Client IP condition to match users who do not have an IP address from the New York office. The Action is set to Redirect with a URL to the resource protected by a smart card contract. The priority of the Rule is set to 2. The rule description indicates that this rule redirects non-headquarter users.	A user, who is at a branch office and is unauthenticated, sees the redirect, is prompted for smart card credentials, and then is granted access to the resource. The page redirects the user to the corporate site.

Figure 28-16 *Rule 2 with Conditions for the Branch Office Users*



Type: Access Gateway: Authorization

Description: Redirect non-headquarter users to smart card authentication

Priority: 2

Conditions

Condition structure: AND Conditions, OR groups

If

☑ Condition Group 1

New

☑ If Not Roles for Current User

Comparison: String : Equals

Mode: Case Sensitive

Value: Roles authenticated

Result on Condition Error: False

☑ And If Not Client IP

Comparison: IP : In Range

Value: Data Entry Field : 10.10.15.1 - 10.10.15.253

Result on Condition Error: False

Actions

Do Redirect Redirect to URL:

http://www.mycompany.com/smartcard.html

Changes made on this panel must be applied from the Policies Panel.

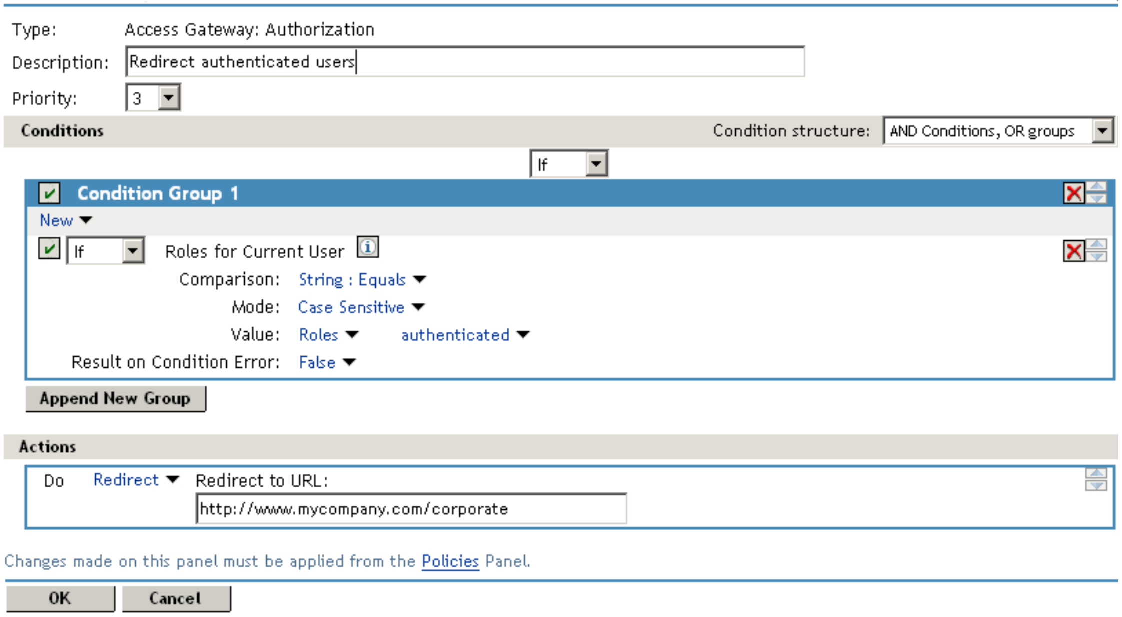
In Rule 2, both conditions are configured to be If Not conditions. Users match the conditions when they are not authenticated and when they do not have a corporate IP address.

Characteristics of Rule 3

You need to configure Rule 3 to have the following conditions, actions, and other characteristics:

Characteristics	User Experience
Condition Group 1 sets the Roles for Current User condition to check for the authenticated role. The Action is set to Redirect with a URL to the corporate site. The priority of the Rule is set to 3. The rule description indicates that this rule redirects authenticated users.	The authenticated user is automatically granted rights to the corporate site if the contract that was used for authentication can satisfy the requirements of the username/password contract.

Figure 28-17 *Rule 3 for Already Authenticated Users*



Type: Access Gateway: Authorization
Description: Redirect authenticated users
Priority: 3
Condition structure: AND Conditions, OR groups
If
If Roles for Current User
Comparison: String : Equals
Mode: Case Sensitive
Value: Roles authenticated
Result on Condition Error: False
Do Redirect Redirect to URL:
http://www.mycompany.com/corporate

Changes made on this panel must be applied from the Policies Panel.

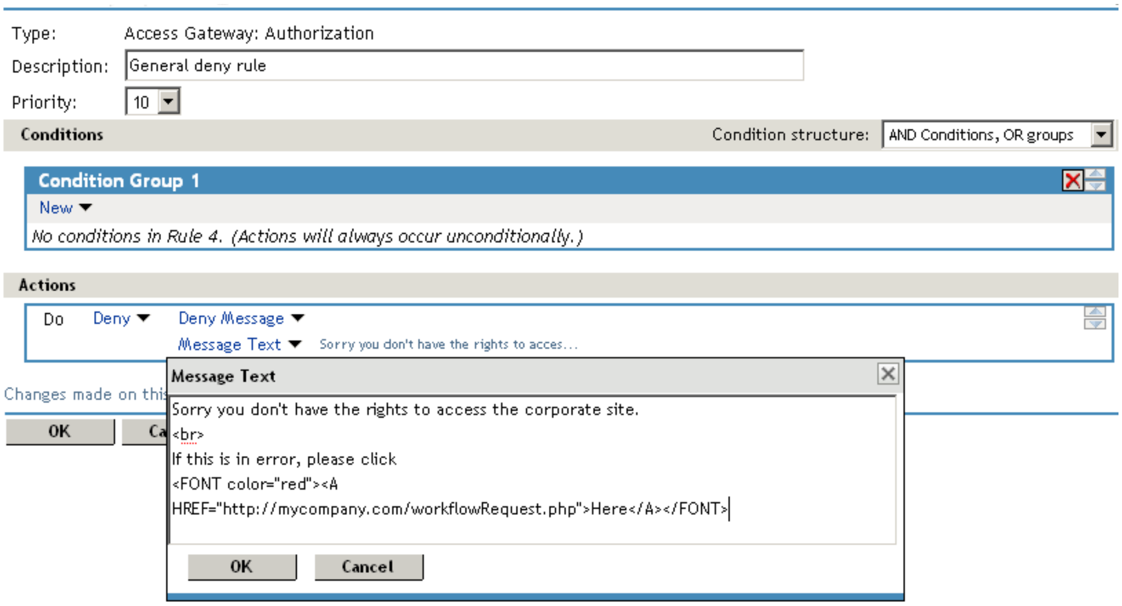
If your branch office users have a method of getting to the Identity Server and logging in with a username/password instead of a smart card contract, you need to modify Rule 3 to accommodate this situation. You need to create two Condition Groups. The first one checks for the authenticated role and the IP address of a corporate user. The second condition group checks for the authenticated role, the IP address of a branch office user, and for the smart card authentication contract. If the user matches either of these Condition Groups, the user is redirected to the corporate site. You need to add another rule to redirect the authenticated branch office users who authenticated with the username/password contract. This rule needs to redirect these users to the site that requires a smart card authentication. This can be the same URL you used for Rule 2.

Characteristics of Rule 4

You need to configure Rule 4 to have the following conditions, actions, and other characteristics:

Characteristics	User Experience
No conditions are specified. The Action is set to Deny. The priority of the Rule is set to 10. The rule description indicates that this rule is a general deny rule.	If the user does not match Rule 1 through 3, the user is denied access.

Figure 28-18 *Rule 4 with a Deny Message*



Type: Access Gateway: Authorization

Description: General deny rule

Priority: 10

Conditions Condition structure: AND Conditions, OR groups

Condition Group 1

No conditions in Rule 4. (Actions will always occur unconditionally.)

Actions

Do Deny Deny Message

Message Text Sorry you don't have the rights to acces...

Changes made on thi

Message Text

```
Sorry you don't have the rights to access the corporate site.
<br>
If this is in error, please click
<FONT color="red"><A
HREF="http://mycompany.com/workflowRequest.php">Here</A></FONT>
```

The deny message in Rule 4 can contain plain text or text with HTML tags.

Other Considerations

If you want to control which resources users are allowed access based on the contract the user used for authentication, you can use the Contract condition to create a Role policy. The role, which is assigned when the user authenticates, can then be used in Authorization policies to permit or deny access.

This scenario only works if your users have IP addresses. If your users are behind a forward proxy, you might want to consider using the X-Forwarded-For IP condition. The forward proxy fills in this HTTP header value. If you know the IP addresses of all the forward proxies at the branch offices, you can use this condition to configure the rules to redirect the users to the appropriate resource and contract for authentication.

28.7 Using Multiple Conditions

The *Condition structure* option controls how conditions within a condition group interact with each other and how condition groups interact with each other. Select one of the following:

- **AND Conditions, OR groups:** If the conditions are ANDed, the user must meet all the conditions in a condition group to match the profile. If the condition groups are ORed, the user must meet all of the conditions of one group to match the profile. This option allows you to set up two or more profiles into which a user could fit and be considered a match. For example, you could create the following Permit rule:

  The first condition group could contain the following conditions:

  1. The user's department must be Engineering.
  2. The request must come on a weekday.

  The second condition group could contain the following conditions:

  1. The user's department must be Information Services and Technology (IS&T).
  2. The request must come on a weekend.

  With this rule, the engineers who match the first condition group have access to the resource during the week, and the IS&T users who match the second condition group have access to the resource on the weekend.

- **OR Conditions, AND groups:** If the conditions are ORed, the user must meet at least one condition in the condition group to match the profile. If the conditions groups are ANDed, the user must meet at least one condition in each condition group to match the profile. For example, suppose you created the following allow rule:

  The first condition group could contain the following conditions:

  1. The user's department is Engineering.
  2. The user's department is Sales.

  The second condition group could contain the following conditions:

  1. The user has been assigned the Party Planning role.
  2. The user has been assigned the Vice President role.

  With this rule, the Vice Presidents of both the Engineering and Sales departments can access the resource, and the users from the Engineering and Sales department who have been assigned to the Party Planning role can access the resource.

At the top of each condition group, there is an option that allows you to control whether the user must match the conditions to match the profile or whether the user matches the profile if the user doesn't match any of the conditions. Depending upon your selection for the Condition structure, you can select from the following:

- If/If Not
- Or/Or Not
- And/And Not

Conditions also have similar Not options, so that a user can match a condition by not matching the specified value.

The check box ☑ by each condition allows you to enable the condition or disable it. You usually disable a condition when testing a new rule, and if you decide the condition is not needed, you can then use the *Delete* ☒ button to delete the condition from the rule. Use the *Move* ⇕ buttons by the *Delete* button to move a condition up or down within its group.

28.8 Importing and Exporting Authorization Policies

You can import and export Authorization policies in order to run them in other Access Manager configurations and to analyze the authorization logic. The policy is exported as a text file with XML tags. We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy ought to be done through the Administration Console.

To export an Authorization policy:

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Select an Authorization policy, then click *Export*.
3. (Optional) Modify the name suggested for the file.
4. Click *OK*
5. Using the features of your browser, specify where the file is copied.

To import a policy:

1. Make sure any referenced Role policies have been imported.

   See Section 27.6, “Importing and Exporting Role Policies,” on page 412.
2. If the policy uses LDAP or Liberty Profile attributes, make sure the Identity Server has been configured for these same attributes.
3. In the Administration Console, click *Policies*.
4. Click *Import*, then browse to the location of the file.
5. Click *OK*.
6. When the policy appears in the list, click *Apply Changes*.

Creating Identity Injection Policies

29

Identity injection allows you to add information to the URL or to the HTML page before it is posted to the Web server. The Web server uses this information to determine whether the user should have access to the resource, so it is the Web server that determines the information that you need to inject to allow access to the resource.

Identity injection is one of the features of Access Manager that enable you to provide single sign-on for your users. When the policy is configured correctly, the user is unaware that additional information is required to access a Web server.

IMPORTANT: Identity Injection policies allow you to inject the user's password into the HTTP header. If you set up such a policy, you should also configure the Access Gateway to use SSL between itself and the back-end Web server. This is the only way to ensure that the password is encrypted on the wire.

This section describes the elements available for an Identity Injection policy, but your Web servers determine which elements you use.

29.1 Designing an Identity Injection Policy

Before setting up an Identity Injection policy, you need to know the following about your Web application:

- Does it require an authentication header? Does this header need just the user name or does it also need the password?
- Does it use a custom header with custom names (x-names)? If so, you need to know their names and their expected values.
- Does the custom header require any custom names (x-names) with tags? If so, gather this information.
- Does the application expect specific values in the query string of the URL? If so, gather this information.

After gathering the information, you need to determine whether you need to create one policy with one rule, one policy with multiple rules, or multiple policies. If you have multiple applications that require the same type of authentication header, you might want to create an authentication header policy and separate policies for the application-specific information. You can then enable both the authentication header policy and the application-specific policy for the resource that is protecting the application. Everything defined in a policy is injected into the header, even if the values are empty because the Access Manager could not obtain the value for the item. For some applications, this is still useful information and the application uses it to make access decisions.

You should design your policies so that the application receives just what it needs. It should not inject custom names and values it does not use.

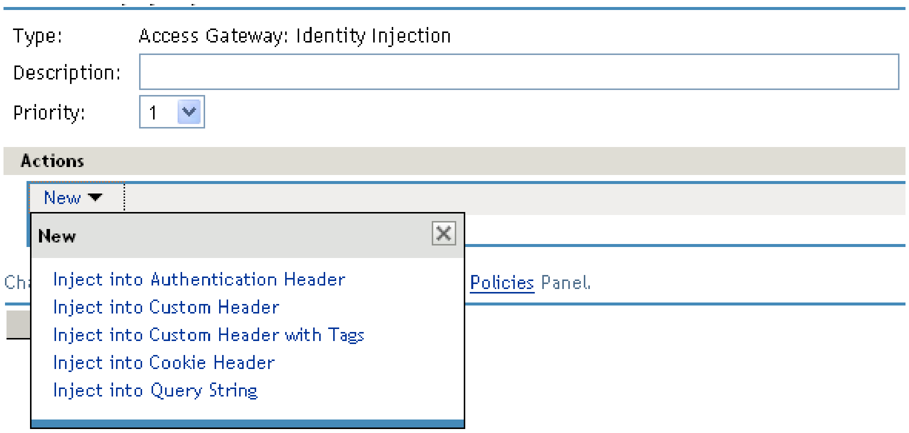
Whether you create a policy with one rule or multiple rules is a personal design decision. If you put all the actions in one rule, you have only one description field to describe the function of the policy. If you put each action type in a separate rule, you have multiple description fields to describe the function of the policy. Select the method that is easiest for you.

Rules are evaluated by priority. The first rule that is evaluated with an authentication header is processed, and the authentication header is rejected if it is found in any of the other rules. Your policy can inject only one authentication header, one cookie header, and one query string, but it can inject multiple custom headers and custom headers with tags.

29.2 Configuring an Identity Injection Policy

1 In the Administration Console, click *Access Manager* > *Policies* > *New*.

2 Specify a name for the policy, then select *Access Gateway: Identity Injection* for the type of policy.



Type: Access Gateway: Identity Injection

Description:

Priority: 1

Actions
New
New
Inject into Authentication Header
Inject into Custom Header
Inject into Custom Header with Tags
Inject into Cookie Header
Inject into Query String
Policies Panel.

3 Fill in the following fields:

Description: (Optional) Describe the purpose of this policy. Because Identity Injection policies are customized to match the content of a specific Web server, you might want to include the name of the Web server as part of the description.

Priority: Specify the order in which a rule is applied in the policy, when the policy has multiple rules. The highest priority is 1 and 10 is the lowest.

4 In the *Actions* section, click *New* and select one of the following.

Repeat this process to add multiple actions to the same rule. If a particular action is allowed only once per rule, then the action does not appear in the *New* menu if that action has already been defined in the rule.

- **Inject into Authentication Header:** Inserts the user name and password into the header. For information about how to configure this type of policy, see Section 29.3, “Configuring an Authentication Header Policy,” on page 469.
- **Inject into Custom Header:** Inserts custom names with values into the custom header. For information about how to configure this type of policy, see Section 29.4, “Configuring a Custom Header Policy,” on page 472.
- **Inject into Custom Header with Tags:** Inserts custom tags with name/value content into the custom header. For information about how to configure this type of policy, see Section 29.5, “Configuring a Custom Header with Tags,” on page 474.
- **Inject into Query String:** Inserts a query string into the URL for the page. For information about how to configure this type of policy, see Section 29.6, “Specifying a Query String for Injection,” on page 476.
- **Inject into Cookie Header:** Inserts the session cookie into the cookie header. For information about how to configure this type of policy, see Section 29.7, “Injecting into the Cookie Header,” on page 478.

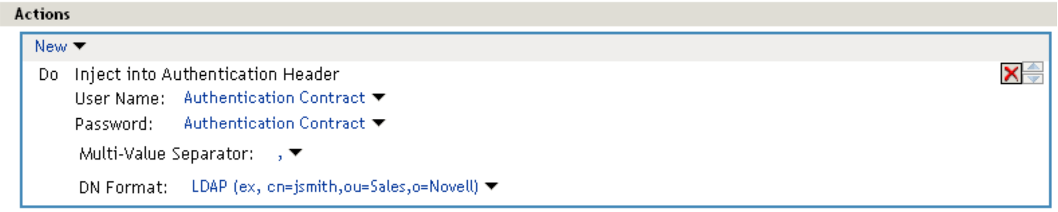
5. (Optional) Repeat Step 4.
6. To save the policy, click *OK* twice, then click *Apply Changes*.
7. For information on how to assign the policy to a protected resource, see Section 13.4.2, “Assigning an Authorization Policy to a Protected Resource,” on page 210.

29.3 Configuring an Authentication Header Policy

To inject values into the authentication header, you need to know what the Web server requires. For basic authentication, you need to inject the user name and password. For a sample policy for a Web server that requires the LDAP username and password to be injected into the header, see “Setting Up an Identity Injection Policy” in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

To create and configure an authentication header policy:

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.
3. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple policies to be used by multiple resources.
4. In the *Actions* section, click *New*, then select *Inject into Authentication Header*.



Do Inject into Authentication Header
User Name: Authentication Contract
Password: Authentication Contract
Multi-Value Separator: ,
DN Format: LDAP (ex, cn=jsmith,ou=Sales,o=Novell)

5 Fill in the *User Name* field.

Select *Credential Profile* to insert the name the user entered when the user authenticated. This is the most common value type to use for user name.

The default contracts assign the cn attribute to the Credential Profile. If you create your own authentication contract, you can assign a different attribute to the Credential Profile.

If your user store is an Active Directory server, you need to be aware that the cn attribute is used even though the user login is chosen from the SAMAccountName attribute. If you want to use the SAMAccountName attribute in the Credential Profile, you need to create your own authentication contract.

Depending upon what the user must supply for authentication, select one of the following:

- **LDAP Credentials:** If you prompt the user for a user name, select this option, then select either *LDAP User Name* (the cn attribute of the user) or *LDAP User DN* (the fully distinguished name of the user). Your Web server requirements determine which one you use.
- **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following. Your Web server requirements determine which one you use.
  - **X509 Public Certificate Subject:** Injects just the subject field from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Injects just the issuer field from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Injects the entire certificate.
  - **X509 Serial Number:** Injects the certificate serial number.
- **SAML Credential:** Although this option is available for the user name, most applications that use SAML assertions use them for the user's password. For the user name, you should probably select an option that allows you to supply the user's name, for example LDAP Credentials or LDAP Attribute.

Your Web server requirements determine the data type you select for the user name. LDAP, X509, and SAML credentials are available from the Credential Profile.You can also select one of the following values to insert into the header as the user name:

- **Authentication Contract:** Injects the URI of the authentication contract the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **LDAP Attribute:** Injects the value of the selected attribute. For Active Directory servers, specify the SAMAccountName attribute for the user name.
- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See Section 12.2, "Enabling Web Services and Profiles," on page 174.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles for Current User:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects the user name that has been stored in the selected shared secret store.

  You can create your own user name attribute. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The store can contain

  one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, “Creating and Managing Shared Secrets,” on page 499.
- **String Constant:** Injects a static value that you specify in the text box. This name is used by all users who access the resources assigned to this policy.
- **Java Data Injection Module:** Specifies the name of a custom Java plug-in, which injects custom values into the header. Usually, you can use either the *LDAP Attribute* or *Liberty User Profile* option to supply custom values, because both are extensible. For more information about creating a custom plug-in, see Novell Access Manager Developer Tools and Examples (http://developer.novell.com/wiki/index.php/Nacm).

The value type you use depends upon how you have set up the application.

6 Fill in the *Password* field.

Select *Credential Profile* to insert the password the user entered when the user authenticated. This is the most common value type to use for the password. Depending upon what the user must supply for authentication, select one of the following:

- **LDAP Credentials:** If you prompt the user for a password, select this option, then select *LDAP Password*. If the user’s password is the same as the name of the user, you can select either *LDAP User Name* (the cn attribute of the user) or *LDAP User DN* (the fully distinguished name of the user).
- **X509 Credentials:** If you use a certificate for the password, select this option, then select one of the following:
  - **X509 Public Certificate Subject:** Injects just the subject from the certificate, which can match the DN of the user, depending upon who issued the certificate.
  - **X509 Public Certificate Issuer:** Injects just the issuer from the certificate, which is the name of the certificate authority (CA) that issued the certificate.
  - **X509 Public Certificate:** Injects the entire certificate.
  - **X509 Serial Number:** Injects the certificate serial number.
- **SAML Credential:** Injects the SAML assertion in the authentication header as the user’s password.

Your Web server requirements determine the data type you select for the password. LDAP, X509, and SAML credentials are available from the Credential Profile. You can also select one of the following values to insert into the header as the password:

- **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **LDAP Attribute:** Injects the value of the selected attribute.
- **Liberty User Profile:** Injects the value of the selected attribute.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles for Current User:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects the password that has been stored in the selected shared secret store.

  You can create your own password attribute. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The store can contain one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, “Creating and Managing Shared Secrets,” on page 499.

- **String Constant:** Injects a static value that you specify in the text box. This name is used by all users who access the resources assigned to this policy.
- **Java Data Injection Module:** Specifies the name of a custom Java plug-in, which injects custom values into the header. Usually, you can use either the *LDAP Attribute* or *Liberty User Profile* option to supply custom values, because both are extensible. For more information about creating a custom plug-in, see *Novell Access Manager Developer Tools and Examples* (http://developer.novell.com/wiki/index.php/Nacm).

The value type you use depends upon how you have set up the application.

7 Specify the format for the value:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, *Roles for Current User* can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed, comma notation: For example:
  `cn=jsmith,ou=Sales,o=novell`
- **NDAP Partial Dot Notation:** Specifies eDirectory™ typeless, dot notation: For example:
  `jsmith.sales.novell`
- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless, leading dot notation.
  `.jsmith.sales.novell`
- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed, dot notation. For example:
  `cn=jsmith.ou=Sales.o=novell`

8 Click *OK*.

9 (Optional) To add a second rule, click *New* in the Rule List.

You can inject only one authentication header into an Identity Injection rule. However, your policy can have multiple rules. If you inject two authentication headers, each in a separate rule, the authentication header in the rule with the highest priority is applied, and the authentication header action in the second rule is ignored.

10 To save the policy, click *OK*, then click *Apply Changes*.

29.4 Configuring a Custom Header Policy

To inject values into a custom header, you need to know the name of the tag and its expected value type. The names are specific to the application. The names might be case sensitive. They might require an X- prefix. Because the requirements vary, you need to enter them in the format as specified by the application. For example, an application might require the following to be in the custom header:

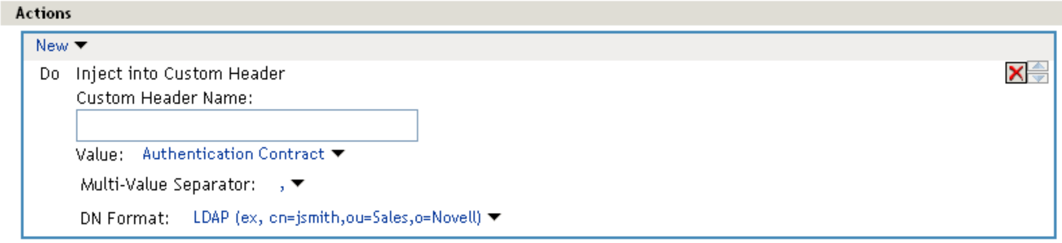
Name/Value Pair	Description
`X-First_Name=givenName`	A first name tag with an LDAP attribute value
`X-Last_Name=sn`	A last name tag with an LDAP attribute value

Name/Value Pair	Description
`X-Role=sales_role`	A role tag with the role name as the value.

If you create a custom header policy with these name/value pairs, the policy injects these names with their values into a custom header, before sending the request to the Web server.

To create such a policy:

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.
3. (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple custom header policies to be used for multiple resources.
4. In the *Actions* section, click *New*, then select *Inject into Custom Header*.



Do Inject into Custom Header
Custom Header Name:
Value: Authentication Contract
Multi-Value Separator: ,
DN Format: LDAP (ex, cn=jsmith,ou=Sales,o=Novell)

5. Fill in the following fields:

   Custom Header Name: Specify the name to be inserted into the custom header. These are the names required by your application. If your application requires the X- prefix, make sure you include the prefix in this field.

   Value: Select the value required by the name. Select one of the following:

   - **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
   - **Client IP:** Injects the IP address associated with the user.
   - **Credential Profile:** Injects the credentials that the user specified at login. You can select *LDAP Credentials*, *X509 Credentials*, or *SAML Credentials*. For more information, see Section 29.3, "Configuring an Authentication Header Policy," on page 469.
   - **LDAP Attribute:** Injects the value of the selected attribute.
   - **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See Section 12.2, "Enabling Web Services and Profiles," on page 174.
   - **Proxy Session Cookie:** Injects the session cookie associated with the user.
   - **Roles for Current User:** Injects the roles that have been assigned to the user.
   - **Shared Secret:** Injects a value that has been stored in the selected shared secret store. Select the shared secret store and the name of the value you want injected.

     You can create your own value. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The name you select for the attribute should match the Custom Header name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.

- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Java Data Injection Module:** Specifies the name of a custom Java plug-in, which injects custom values into the header. Usually, you can use either the *LDAP Attribute* or *Liberty User Profile* option to supply custom values, because both are extensible. For more information, see *Novell Access Manager Developer Tools and Examples* (http://developer.novell.com/wiki/index.php/Nacm).

6 Specify the format for the value:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, *Roles for Current User* can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed, comma notation: For example:
  `cn=jsmith,ou=Sales,o=novell`
- **NDAP Partial Dot Notation:** Specifies eDirectory typeless, dot notation: For example:
  `jsmith.sales.novell`
- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless, leading dot notation.
  `.jsmith.sales.novell`
- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed, dot notation. For example:
  `cn=jsmith.ou=Sales.o=novell`

7 To save the policy, click *OK* twice, then click *Apply Changes*.

29.5 Configuring a Custom Header with Tags

Some Web applications require more than a name and a value to be injected into the custom header. Sometimes they require a custom name, a tag, and a value. Sometimes the application requires a custom name with multiple tags and values. The *Inject into Custom Header with Tags* option provides you with the flexibility to add such values to the custom header. For example, your application could be expecting the following custom header with tag:

```
X-Custom_Role Role=Manager
```

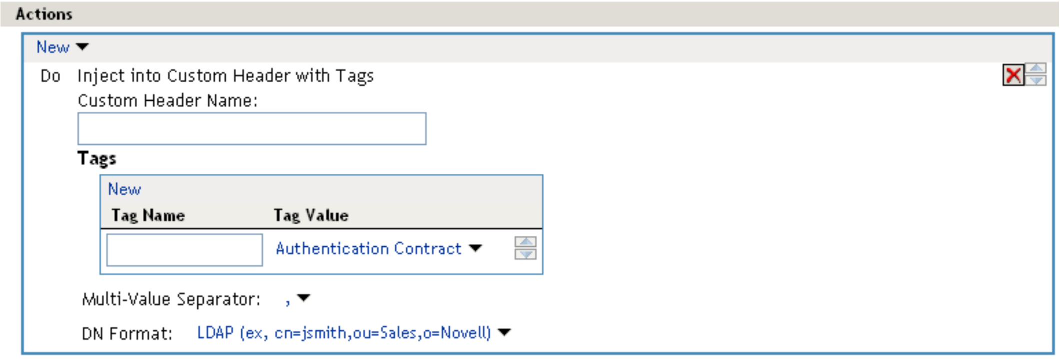
You can inject this information by setting the *Custom Header Name* to X-Custom, the *Tag Name* to Role, and the *Tag Value* to Manager. The value can be set as a static variable or you can retrieve it from various sources such as a Liberty User Profile attribute or the roles assigned to the current user.

1 In the Administration Console, click *Access Manager* > *Policies* > *New*.

2 Specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.

3 (Optional) Specify a description for the injection policy. This is useful if you plan to create multiple custom header policies to be used for multiple resources.

4 In the *Actions* section, click *New*, then select *Inject into Custom Header with Tags*.



5. Fill in the following fields:

   Custom Header Name: Specify the name that the application expects. If your application requires the X- prefix, make sure you include the prefix in this field.

   Tag Name: Specify the tag name that the application expects.

   Tag Value: Specify the value. Select from the following data types:

   - **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
   - **Client IP:** Injects the IP address associated with the user.
   - **Credential Profile:** Injects the credentials that the user specified at login. You can select *LDAP Credentials, X509 Credentials*, or *SAML Credential*. For more information, see Section 29.3, "Configuring an Authentication Header Policy," on page 469.
   - **LDAP Attribute:** Injects the value of the selected attribute.
   - **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See Section 12.2, "Enabling Web Services and Profiles," on page 174.
   - **Proxy Session Cookie:** Injects the session cookie associated with the user.
   - **Roles for Current User:** Injects the roles that have been assigned to the user.
   - **Shared Secret:** Injects a value that has been stored in the selected shared secret store. The name specified as the Tag Name must match the name of a name/value pair stored in the shared secret.

     You can create your own value. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The name must match the expected Tag Name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.
   - **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
   - **Java Data Injection Module:** Specifies the name of a custom Java plug-in, which injects custom values into the header. Usually, you can use either the *LDAP Attribute* or *Liberty User Profile* option to supply custom values, because both are extensible. For more information about creating a custom plug-in, see *Novell Access Manager Developer Tools and Examples* (http://developer.novell.com/wiki/index.php/Nacm).
6. To add multiple tag and value pairs to the custom name, click *New* in the *Tags* section.

Use the up-arrow and down-arrow buttons to order the tags.

7. Specify the format for the value:

   Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, *Roles for Current User* can contain multiple values.

   DN Format: If the value is a DN, select the format for the DN:

   - **LDAP:** Specifies LDAP typed, comma notation: For example:

     `cn=jsmith,ou=Sales,o=novell`
   - **NDAP Partial Dot Notation:** Specifies eDirectory typeless, dot notation: For example:

     `jsmith.sales.novell`
   - **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless, leading dot notation.

     `.jsmith.sales.novell`
   - **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed, dot notation. For example:

     `cn=jsmith.ou=Sales.o=novell`

8. To save the policy, click *OK* twice, then click *Apply Changes*.

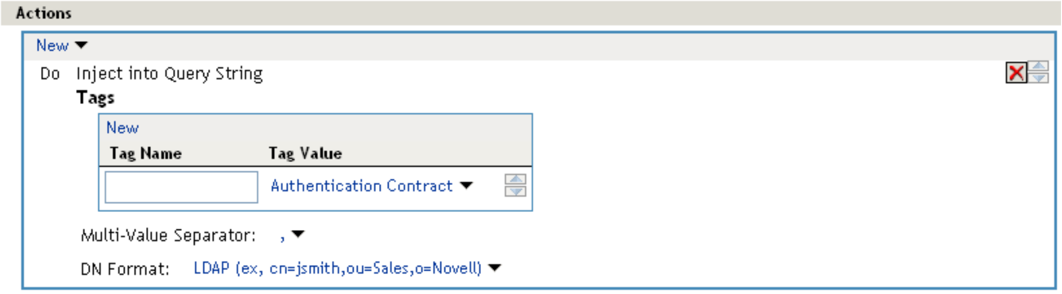
29.6 Specifying a Query String for Injection

Some applications require custom information in a query string of the URL. The *Inject into Query String* option allows you to inject this information without prompting the user for it. To inject the information, you must specify a tag name and a tag value. The tag name is what your application requires. For example, suppose your application expects the following query string for user jsmith:

`?name=jsmith`

You can inject this information into the URL by specifying a name for the *Tag Name* and *Credential Profile* for the *Tag Value*. The *Credential Profile* value type inserts the name the current user specified when authenticating to the Access Gateway.

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.
3. (Optional) Specify a description for the injection policy.
4. In the *Actions* section, click *New*, then select *Inject into Query String*.



Do Inject into Query String

Tag Name	Tag Value
	Authentication Contract

Multi-Value Separator: ,

DN Format: LDAP (ex, cn=jsmith,ou=Sales,o=Novell)

5. Fill in the following fields:

   Tag Name: Specify the tag name that the application expects.

Tag Value: Specify the value. Select from the following data types:

- **Authentication Contract:** Injects the URI of a local authentication contract that the user used for authentication.
- **Client IP:** Injects the IP address associated with the user.
- **Credential Profile:** Injects the credentials that the user specified at login. You can select *LDAP Credentials, X509 Credentials*, or *SAML Credential*. For more information, see Section 29.3, "Configuring an Authentication Header Policy," on page 469.
- **LDAP Attribute:** Injects the value of the selected attribute.
- **Liberty User Profile:** Injects the value of the selected attribute. If no profile attributes are available, you have not enabled their use in the Identity Server configuration. See Section 12.2, "Enabling Web Services and Profiles," on page 174.
- **Proxy Session Cookie:** Injects the session cookie associated with the user.
- **Roles for Current User:** Injects the roles that have been assigned to the user.
- **Shared Secret:** Injects a value that has been stored in the selected shared secret store. The name specified as the Tag Name must match the name of a name/value pair stored in the shared secret.

  You can create your own value. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The name you specify must match the Tag Name. The store can contain one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.
- **String Constant:** Injects a static value that you specify in the text box. This value is used by all users who access the resources assigned to this policy.
- **Java Data Injection Module:** Specifies the name of a custom Java plug-in, which injects custom values into the header. Usually, you can use either the *LDAP Attribute* or *Liberty User Profile* option to supply custom values, because both are extensible. For more information about creating a custom plug-in, see *Novell Access Manager Developer Tools and Examples* (http://developer.novell.com/wiki/index.php/Nacm).

6 (Optional) To add multiple tag and value pairs, click *New* in the *Tags* section.

You can inject only one query string into a rule, but you can inject multiple tag-name and tag-value pairs in the single query string.

Use the up-arrow and down-arrow buttons to order the tags.

7 Specify the format for the values:

Multi-Value Separator: Select a value separator, if the value type you have select is multi-valued. For example, *Roles for Current User* can contain multiple values.

DN Format: If the value is a DN, select the format for the DN:

- **LDAP:** Specifies LDAP typed, comma notation: For example:

  `cn=jsmith,ou=Sales,o=novell`
- **NDAP Partial Dot Notation:** Specifies eDirectory typeless, dot notation: For example:

  `jsmith.sales.novell`
- **NDAP Leading Partial Dot Notation:** Specifies eDirectory typeless, leading dot notation.

  `.jsmith.sales.novell`

- **NDAP Fully Qualified Partial Dot Notation:** Indicates eDirectory typed, dot notation. For example:

  `cn=jsmith.ou=Sales.o=novell`

8 To save the policy, click *OK* twice, then click *Apply Changes*.

29.7 Injecting into the Cookie Header

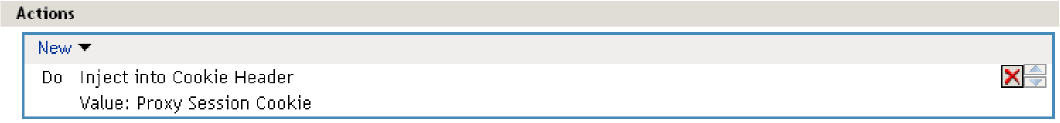
Some applications require access to the Access Gateway session cookie and expect to find it in the cookie header. You can create an Identity Injection policy that adds this cookie to the cookie header.

1 In the Administration Console, click *Access Manager* > *Policies* > *New*.

2 Specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.

3 (Optional) Specify a description for the injection policy.

4 In the *Actions* section, click *New*, then select *Inject into Cookie Header*.



Do Inject into Cookie Header
Value: Proxy Session Cookie

This action allows only one value, so the value is configured automatically.

5 To save the policy, click *OK* twice, then click *Apply Changes*.

29.8 Importing and Exporting Identity Injection Policies

You can import and export Identity Injection policies in order to run them in other Access Manager configurations. The policy is exported as a text file with XML tags. We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy should to be done through the Administration Console.

To export an Identity Injection policy:

1 In the Administration Console, click *Access Manager* > *Policies*.

2 Select an Identity Injection policy, and click *Export*.

3 (Optional) Modify the name suggested for the file.

4 Click *OK*.

5 Using the features of your browser, specify where the file is to be copied.

To import a policy:

1 Make sure any referenced shared secret stores have been created. See Section 30.4, "Creating and Managing Shared Secrets," on page 499.

2 If the policy uses LDAP or Liberty Profile attributes, make sure the Identity Server has been configured for these same attributes.

3 Make sure any referenced role policies have been imported.

   See Section 27.6, "Importing and Exporting Role Policies," on page 412.

4 In the Administration Console, click *Access Manager* > *Policies*.

5 Click *Import*, then browse to the location of the file.

6 Click *OK*.

7 When the policy appears in the list, click *Apply Changes*.

29.9 Sample Identity Injection Policy

One of the common uses of an Identity Injection policy is to differentiate between internal users and external users. Web servers that have been configured for this logic can then display one set of pages to internal users and another set of pages to external users. The following sample policy is based on such an environment, which has the following characteristics:

- The Web server has been configured to look for a custom tag called `IPAddress` and to differentiate between internal IP addresses and external IP addresses.
- The internal customers have NAT IP addresses.
- The protected resource is a page called `mycompany.html`. This page is a public protected resource (no authentication required) because the IP address of the client is available before authentication.

To configure your site for this type of policy:

1 In the Administration Console, click *Access Manager* > *Policies*.

2 Click *New*, specify a name for the policy, select *Access Gateway: Identity Injection* for the type, then click *OK*.

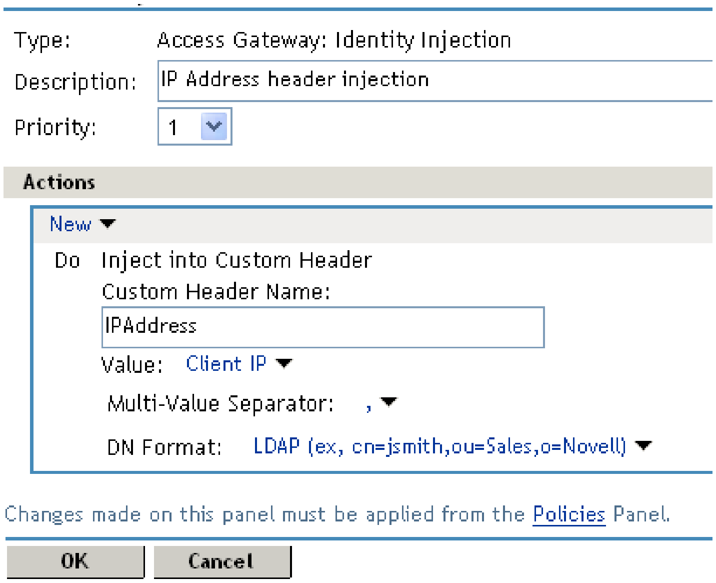
3 In the *Actions* section, click *New* > *Inject into Custom Header*.

4 Fill in the following fields:

Custom Header Name: Specify `IPAddress` in the text box.

Value: Select *Client IP*.

The other fields do not need to be modified. Your policy should look similar to the following:



Type: Access Gateway: Identity Injection

Description: IP Address header injection

Priority: 1

Actions

New

Do Inject into Custom Header

Custom Header Name: IPAddress

Value: Client IP

Multi-Value Separator: ,

DN Format: LDAP (ex, cn=jsmith,ou=Sales,o=Novell)

Changes made on this panel must be applied from the Policies Panel.

5. Click *OK* twice, then click *Apply Changes*.
6. Assign the policy to the `mycompany.html` page of the Web server. Click *Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > Protected Resources*.
7. In the Protected Resource List, select the protected resource for the page or click *New* to create one, then specify a name for it.
8. In the *URL Path List*, ensure that the path ends with the name of the page. For example:

   `/mycompany.html`
9. Click *Identity Injection*, select the name of the IP address policy, then click *Enable*.
10. To save the changes, click *Configuration Panel > OK*.
11. On the Configuration page, click *OK*, then click *Update*.
12. Configure the Web server to use the IPAddress values in the custom header to distinguish between external and internal customers.

    In this sample scenario, the Web server is configured to recognize IP addresses starting with `10.` as internal customers and all other addresses as external customers.

Creating Form Fill Policies

30

A Form Fill policy allows you to prepopulate fields in a form on first login and then save the information in the completed form to a secret store for subsequent logins. The user is prompted to reenter the information only when something changes such as an expired password. Form Fill is one of the features of Access Manager that enable you to provide single sign-on for your users.

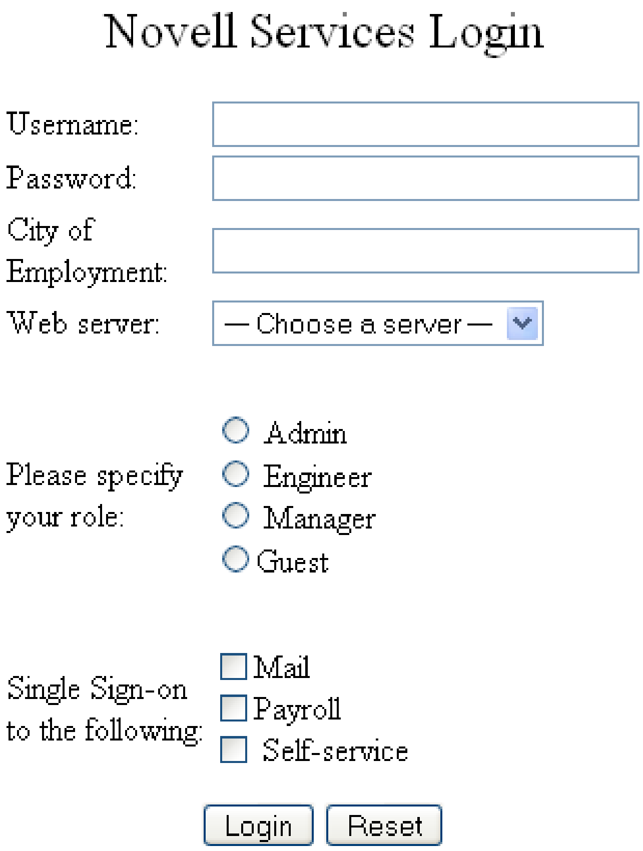
The HTML page determines the requirements for the Form Fill policy. This section describes the following:

- Section 30.1, "Understanding an HTML Form," on page 481
- Section 30.2, "Creating a Form Fill Policy for the Sample Form," on page 484
- Section 30.3, "Implementing Form Fill Policies," on page 487
- Section 30.4, "Creating and Managing Shared Secrets," on page 499
- Section 30.5, "Importing and Exporting Form Fill Policies," on page 501

30.1 Understanding an HTML Form

The following figure is an example of a Web page containing an HTML form.

Figure 30-1 *Sample HTML Form*



The information in this section uses this sample form to explain how to create a policy. This sample form deliberately contains a variety of field types:

- Input items for Username and Password
- Selection options for the Web server field
- Radio buttons for the role
- Check boxes for single sign-on

When analyzing a form, you need to decide if you want the policy to fill in all the fields or just some of them. You then need to look at the source HTML of the form to discover the names of the fields and their types.

An HTML form is created using a set of HTML tags. A form consists of elements (fields, menus, check boxes, radio buttons, push buttons, etc.) that control how the form is completed and submitted. For more detailed information about forms, see the Forms section at www.w3.org (http://www.w3.org/TR/html401/interact/forms.html).

The following HTML data corresponds to the sample form (see Figure 30-1). The lines that contain the information needed to create a Form Fill policy appear in bold type. Each line corresponds to a field in the form that requires information or allows the user to select information.

In the example, each bold line contains information about a field, its name, and type. You use this information in the policy to specify how the information in the field is filled.

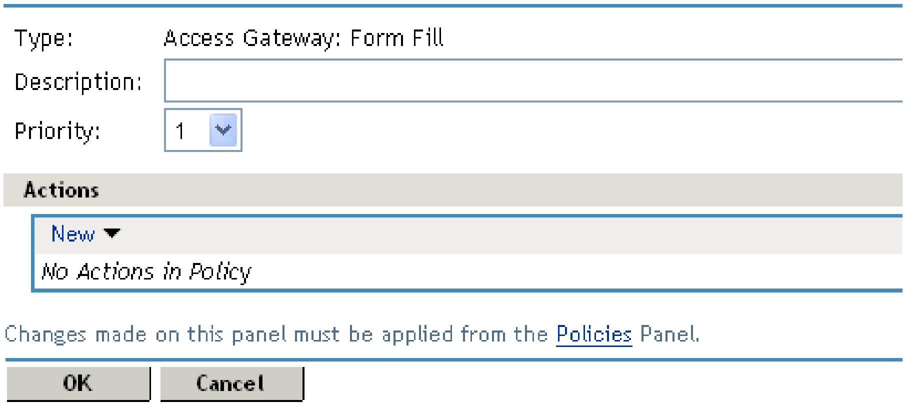
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Form Fill Test Page</title>
</head>
<body>
  <form name="mylogin" action="validatepassword.php" method="post"
        id="mylogin">
    <table align="center" border="0" cellpadding="4" cellspacing="4">
      <tr align="center" valign="top">
        <td>
          <p align="center"><font size="5">Novell Services Login
             </font></p>
          <table align="center" border="0">

            <tr align="left">
              <td>Username:</td>
              <td><input type="text" name="username" size="30"></td>
            </tr>

            <tr align="left">
              <td>Password:</td>
              <td><input type="password" name="password" size="30">
              </td>
            </tr>

            <tr align="left">
              <td>City of<br>Employment:</td>
              <td><input type="text" name="city" size="30"></td>
            </tr>

            <tr align="left">
              <td>Web server:</td>
              <td>
                <select name="webserv" size="1">
                  <option value="default" selected>
                    --- Choose a server ---
                  </option>
                  <option value="Human Resources">
                    Human Resources
                  </option>
                  <option value="Development">
                    Development
                  </option>
                  <option value="Accounting">
                    Accounting
                  </option>
                  <option value="Sales">
                    Sales
                  </option>
                </select>
              </td>
            </tr>

            <tr>
              <td colspan="2" align="left" height="25" valign="top">
                <p></p>
              </td>
            </tr>

            <tr align="left">
              <td>Please specify<br>your role:</td>
              <td>
                 <input name="role" value="admin" type="radio">
                        Admin<br>
                 <input name="role" value="engineer" type="radio">
                       Engineer<br>
                 <input name="role" value="manager" type="radio">
                       Manager<br>
                 <input name="role" value="guest" type="radio">Guest
              </td>
            </tr>

            <tr>
              <td colspan="2" align="left" height="25" valign="top"
                  width="121">
                <p></p>
              </td>
            </tr>

            <tr align="left">
```

```
              <td>Single Sign-on<br>to the following:</td>
              <td>
                <input name="mail" type="checkbox">Mail<br>
                <input name="payroll" type="checkbox">Payroll<br>
                <input name="selfservice" type="checkbox">
                             Self-service<br>
              </td>
            </tr>
          </table>
        </td>
      </tr>

      <tr>
        <td colspan="2" align="center">
          <input value="Login" type="submit">
          <input type="reset">
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

30.2 Creating a Form Fill Policy for the Sample Form

The sample form has ten input fields and five select options that need to be configured in the Form Fill policy. The following steps explain how to create a shared secret to store the values and use that shared secret to create a Form Fill policy for this sample form.

1. To create the policy, click *Policies* > *New*.
2. Specify a display name for the policy and select *Access Gateway: Form Fill* for its type.



Type: Access Gateway: Form Fill

Description:

Priority: 1

Actions

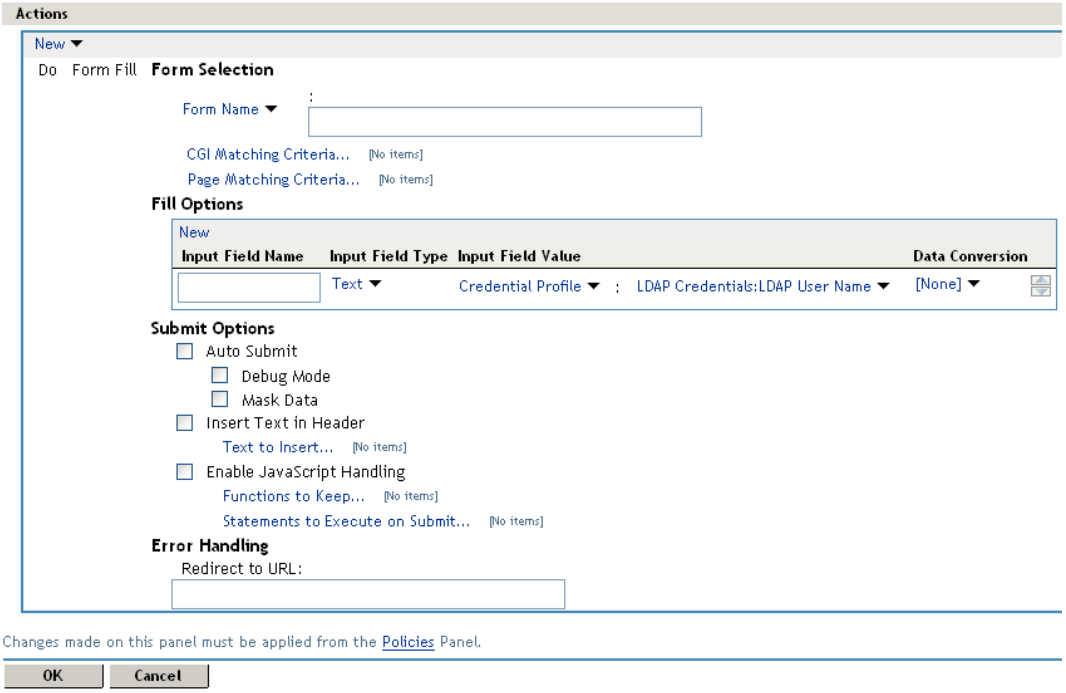
No Actions in Policy

Changes made on this panel must be applied from the Policies Panel.

3. (Optional) Specify a description for the Form Fill policy. This is useful if you plan to create multiple Form Fill policies.

   You might want to specify the name of the HTML page that contains the form this policy is designed to fill.

4 In the *Actions* section, click *New,* then select *Form Fill*.



Changes made on this panel must be applied from the Policies Panel.

5 In the *Form Selection* section, select *Form Name* and specify *mylogin* in the text box. The form name comes from the HTML page. See the following line in the source for the page:

```
<form name="mylogin" action="validatepassword.php" method="post"
        id="mylogin">
```

6 In the *Fill Options* section, specify all the input fields and select options. For each new field, click *New*. Specify the fields in the order in which they appear on the form. The following table displays the Fill Options selected for each input field.

Form Name	Fill Options
username	**Input Field Name:** username **Input Field Type:** Text **Input Field Value:** Credential Profile: LDAP Credentials: LDAP User Name The default contracts assign the cn attribute to the Credential Profile. If you create your own authentication contract, you can assign a different attribute to the Credential Profile. If your user store is an Active Directory server, you need to be aware that the cn attribute is used even though the user login is chosen from the SAMAccountName attribute. If you want to use the SAMAccountName attribute in the Credential Profile, you need to create your own authentication contract.

Form Name	Fill Options
password	**Input Field Name:** password **Input Field Type:** Password **Input Field Value:** Credential Profile: LDAP Credentials: LDAP Password
webserv	**Input Field Name:** webserv **Input Field Type:** Select **Input Field Value:** Shared Secret: sampleLogin: webserv
role	**Input Field Name:** role **Input Field Type:** Radio Button **Input Field Value:** Shared Secret: sampleLogin: role
mail	**Input Field Name:** mail **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: mail
payroll	**Input Field Name:** payroll **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: payroll
selfservice	**Input Field Name:** selfservice **Input Field Type:** Checkbox **Input Field Value:** Shared Secret: sampleLogin: selfservice

7 In the *Submit Options* section, fill in the following fields:

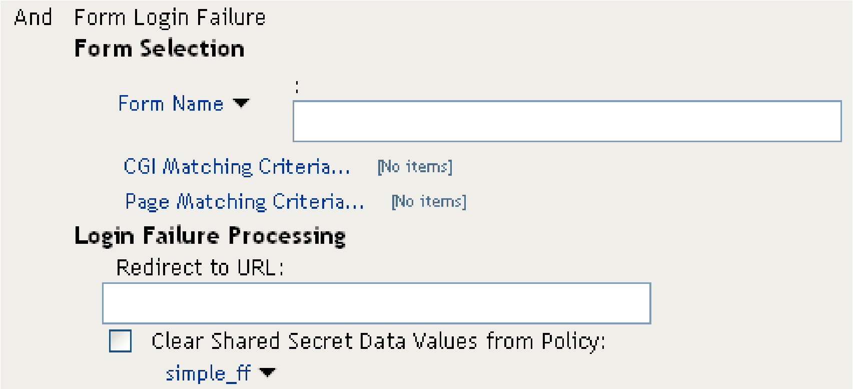
Auto Submit: Select this option to submit the form as soon as all the values are filled in. If this option is not selected, even though all the values are filled in for the user, the user must click the *Submit* button.

Debug Mode: Select the *Debug Mode* option, which allows you to verify that the information is correct before submitting the form. If values must be filled in, you first see the form to add the values. When the form is submitted, you are presented with a JavaScript that contains all of the name/value pairs. To submit the form, you need to click the *Submit* button.

Insert Text in Header: Select this option so you can add a static value. In the *Text to Insert* box, specify the city value. Enter:

```
city = Provo
```

8 To create a login failure policy, click *New* in the *Actions* section, then select *Form Login Failure*.



And Form Login Failure

Form Selection

Form Name:

CGI Matching Criteria... [No items]

Page Matching Criteria... [No items]

Login Failure Processing

Redirect to URL:

☐ Clear Shared Secret Data Values from Policy:

simple_ff

9. In the *Form Selection* section, select *Form Name* and specify *mylogin* in the text box. The form name comes from the HTML page.
10. In the *Login Failure Processing* section, fill in the following field:

    Clear Shared Secret Data Values from Policy: Select this option to clear the data stored in the Shared Secret object when log in fails. Select the name you have given to this policy.
11. Use the up-arrow button to move the Form Login Failure policy to the top of the policy.

    You want the failure policy to execute first on login failure.
12. Click *OK*.
13. On the Policies page, click *Apply Changes*.

30.3 Implementing Form Fill Policies

Section 30.2, "Creating a Form Fill Policy for the Sample Form," on page 484 section describes how to create a simple Form Fill policy for a few input fields. This section describes all available options and explains how to use them to create a Form Fill policy and a Login Failure policy.

- Section 30.3.1, "Designing a Form Fill Policy," on page 487
- Section 30.3.2, "Creating a Form Fill Policy," on page 492
- Section 30.3.3, "Creating a Login Failure Policy," on page 497
- Section 30.3.4, "Troubleshooting a Form Fill Policy," on page 498

30.3.1 Designing a Form Fill Policy

Besides analyzing the form and determining the data items that need to be filled (see Section 30.1, "Understanding an HTML Form," on page 481), you need to consider the following when designing the Form Fill policy:

- "Verifying the Content or Page Type of the Form" on page 487
- "Creating a Form Matching Rule" on page 488
- "Including JavaScript in a Form Fill Policy" on page 490

Verifying the Content or Page Type of the Form

If possible, the URL of the protected resource should include the filename of the page that contains the form. Sometimes this is not possible. If the URL references a directory, the Access Gateway has to sort through the files in the directory and determine which one contains the form. The Linux Access Gateway and the NetWare Access Gateway use different methods to sort through the files.

Linux Access Gateway

The Web server must sent the content type of the form. The Linux Access Gateway processes pages with the following content types:

```
text/html
text/xml
text/css
text/javascript
application/javascript
application/x-javascript
```

If the page with the form has no content type or has a type other than one in the above list, the Linux Access Gateway skips the page.

NetWare Access Gateway

The NetWare Access Gateway looks for the following extensions and excludes these files:

```
gif, jpg, jpeg, pdf, png, zip, jar, bmp, iso, ico, doc, mov, mp3, mpeg,
ppt, rpm, tar, wav, sxi, xls, wmf, wpd, sxw, gz, css, odt
```

If the file has an extension other than one in the above list, the NetWare Access Gateway processes the page.

Creating a Form Matching Rule

To create a successful Form Fill policy, you need to create a matching rule that matches the policy to the HTML page that contains the form, and then matches the form on the page. The Access Gateway uses the following rules, in the order listed, when determining whether a page contains the required form:

1. Matches the protected resource path in the URL with the page. If they don’t match, the page is rejected. If they match, continues. For more information, see “Using the URL of the Protected Resource” on page 489.
2. Checks for CGI criteria. If they don’t match, the page is rejected. If they match or no criteria is specified, continues. For more information, see “Using CGI Matching Criteria” on page 489.
3. Checks for page matching criteria. If they don’t match, the page is rejected. If they match or no page matching criteria is specified, continues. For more information, see “Using Page Matching Criteria” on page 489.
4. Checks the form name criteria (which can be the `<FORM>` name attribute, the `<FORM>` ID attribute, or a number). If it doesn’t match, the page is rejected. If it matches, the form is processed. For more information, see “Using Form Name Criteria” on page 490.

When the Access Gateway uses URL or CGI criteria, it can make a match early in the filling process. This allows the Access Gateway to fill the data from the Web server and send it, almost simultaneously, to the browser. However, if the Access Gateway is configured to use page matching criteria, the Access Gateway must retrieve the entire page from the Web server, process it, and then determine whether the page needs to fill a form. All this processing must be completed before the Access Gateway can send any data to the browser. Unless the page is quite small, users will clearly perceive the delay.

The form name matching criteria are not used for page matching. They are used to determine which form on the page is selected.

Using the URL of the Protected Resource

When assigning a Form Fill policy to a protected resource, we recommend that the URL specified in the *URL Path List* contain the filename of the page. Usually, such a URL is enough to match the HTML page for the form. However, when pages are dynamically generated, sometimes the same filename is used to display different pages. Sometimes you can't specify the filename in the URL. When this is the case, you need to use either the *CGI Matching Criteria* or the *Page Matching Criteria* to create an accurate page matching rule.

Using CGI Matching Criteria

If the page for the URL changes with the CGI portion of the URL (the portion that follows the question mark (?) and also called the query string), you can enter the CGI value. For example, consider the following URL:

```
http://webaccess.novell.com/servlet/webacc?Action=User.logout
```

If this is your URL, you can enter `Action=User.logout` as the value in the text box for the *CGI Matching Criteria* option. If the page generated from this URL always contains the page you want to match, you do not need to add any additional page matching criteria.

Using Page Matching Criteria

If your URL of your protected resource has the following characteristics, you need to use page matching criteria:

- The URL does not contain any CGI data.
- The URL displays generated pages which vary in content. For example, if your form fill login page and the login failure page share the same URL, you need to use page matching criteria.

Page matching criteria are the most processing-intense form of matching and should be avoided if possible, but sometimes they are the only method available to identify the page with the correct form. For example, suppose you have a login failure page and login page that use the same URL, with no CGI data. You can use page matching criteria to ensure that the Access Gateway matches the Form Fill policies for login and for login failure to the correct pages. You need to examine the source code for each page, and identify a string at the top of the page that uniquely identifies the page.

For example, the login page might contain a `<TITLE>` element that names the application the user is logging in to. If the login failure page does not contain the same `<TITLE>` element, you can use the `<TITLE>` element to identity the login page. Suppose the this is true and the login page contains the following string:

```
<TITLE>Novell WebAccess</TITLE>
```

You would add this string as the value in the text box for the *Page Matching Criteria* option. Remember that white space is significant when white space is entered to the left of the value in the text box. To have the Access Gateway ignore white space, left-justify the value in the text box, or copy and paste the HTML text directly from the source code of the Web page.

Now you need to uniquely identify the login failure page. If this page does not have a <TITLE> element, look at the strings near the top of the page. Suppose the page contains the following string:

```
"Please log in again. You might have typed your name or password
incorrectly."
```

Because the login page does not contain this string, you can use this string to identify the login failure page. You would add the following string as the value in the text box for the *Page Matching Criteria* option for the login failure Form Fill policy.

```
Please log in again.
```

To have the Access Gateway ignore white space, left-justify the value in the text box, or copy and paste the HTML text directly from the source code of the Web page.

Using Form Name Criteria

After identifying the page, the Access Gateway needs to identify the form on the page. If there is only one form on the HTML page, the Access Gateway can easily identify the form. If the form has a name or an ID attribute, you can use the value of the attribute to identify the form. If the form doesn't have either of these attributes, you can use the *Number* option with a value of 1. The first form the Access Gateway finds on the page matches.

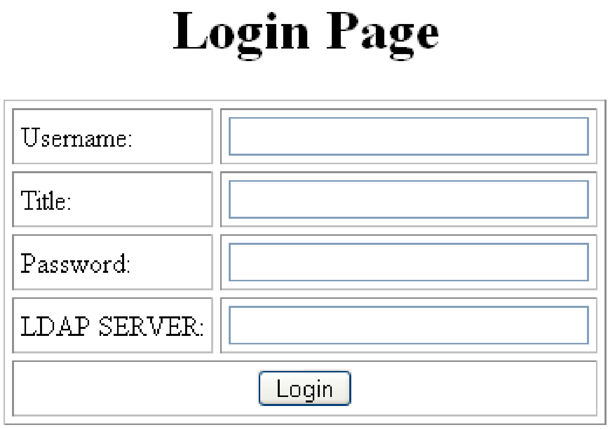
When multiple forms exist on the same HTML page, the easiest and fastest matching method is to give each form a unique name or unique ID on the HTML page. If the forms have the same name or ID, you need to use the Number option, and the order in which they appear on the page determines their number.

The value 0 for the *Number* option has special meaning. You use this value when you want the Form Fill policy to fill in values for all forms on the page. Sometimes a page has multiple forms, but all forms on the page must be filled in before the page can be submitted. For example, one form might contains user information and another form contain user preferences. If both of these forms need to be filled in before the user can log in, then you can use the Number option set to 0, and the Fill Options section of the policy can contain fields for both forms, in the order in which they appear on the page.

Including JavaScript in a Form Fill Policy

Figure 30-2 illustrates a simple form.

Figure 30-2 *Form Login Page*



The source code for this simple form reveals that it includes JavaScript functions:

```
<html><head><title>Login Page</title></head><body>
<h1 align="center">Login Page</h1>
<script language="JavaScript">
      function setCookie(){
            document.cookie="myCookieName=myCookieValue";
      }
      function validate(){
      if(document.mylogin.title.ldap.length == 0){
            alert("You must provide the title for the user!");
            return false;
      }
      return true;
}
</script>
<form name="jscript" action="viewInfo.php" method="post"
onload="setCookie()">
<center>
<table border="1" cellpadding="4" cellspacing="4">
  <tbody><tr>
      <td>Username:</td>
      <td><input name="username" size="30" type="text"></td>
  </tr>

  <tr>
      <td>Title:</td>
      <td><input name="title" size="30" type="text"></td>
  </tr>
  <tr>
      <td>Password:</td>
      <td><input name="password" size="30" type="text"></td>
  </tr>

  <tr>
      <td>LDAP SERVER:</td>
      <td><input name="ldap" size="30" type="text"></td>
  </tr>
  <tr>
      <td colspan="2" align="center">
      <input value="Login" onclick="return validate();" type="submit">
      </td>

  </tr>
</tbody></table>
</center>
</form>

<script language="JavaScript">
function doCookie(){
document.cookie="myCookieName=myCookieValue";
}
return true;
}
</script>
```

```
</body></html>
```

The significant code snippets for determining whether to include JavaScript commands in the Form Fill policy are displayed in bold. The `<script>` elements are in bold because you need to be aware of all the JavaScript on the HTML page. Whether all the functions in the JavaScript need to be included in the policy is usually determined by trial and error. There are some clues you can use to determine the requirements:

- If a function is called within the form, you should include it in the Form Fill policy. The above form calls two JavaScript functions, `setCookie()` and `validate()`.
- If a function is not called by the form, you probably do not need to include it. The above form has one JavaScript function that falls within this category, `doCookie`. You can probably leave out these types of functions, but only trial and error can determine whether that is true.

For this form, you could select the *Enable JavaScript Handling* option. This would include all three functions (`setCookie()`, `validate()`, and `doCookie()`) in the Form Fill policy. If you wanted to test whether the `doCookie()` function was needed, you would select the *Enable JavaScript Handling* option and then specify the following in the *Functions to Keep* text box:

```
function setCookie()
function validate()
```

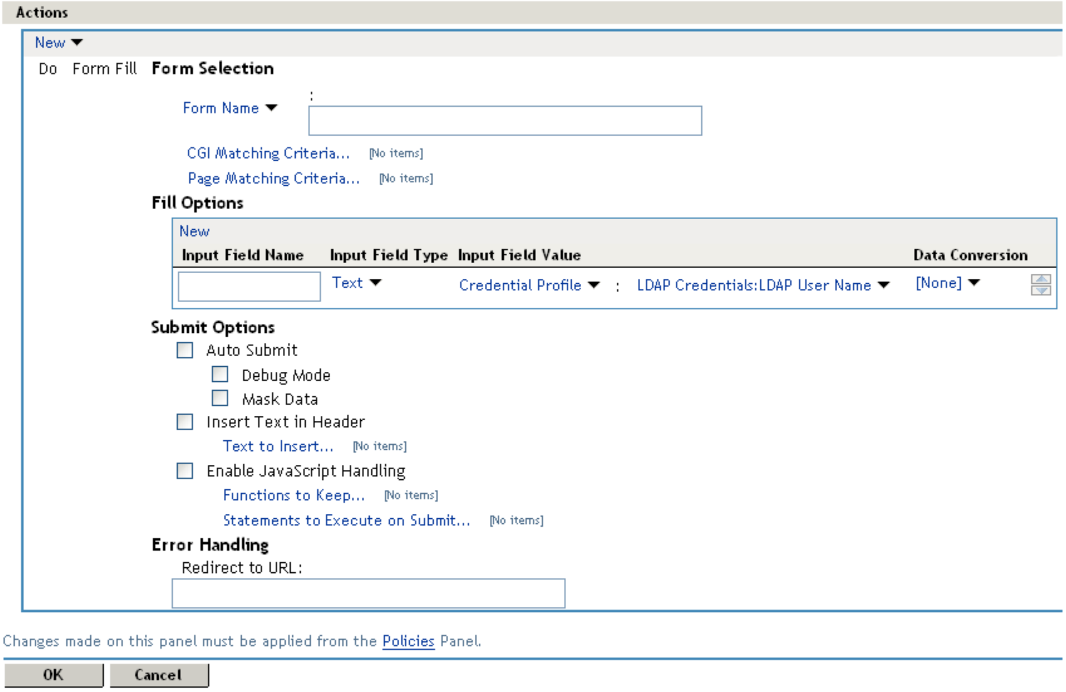
Each function needs to be placed on a separate line.

30.3.2 Creating a Form Fill Policy

1. Examine the source code for the HTML form and determine what data the form requires and where that data is stored (LDAP attributes, Liberty User Profile attributes, shared secrets, credential profiles, etc.)

   Ideally, the form should be its own HTML page, and page should be as small as possible. Form Fill must parse the entire file and assemble the body in contiguous memory before the first byte of the form is displayed to the user. On a large file, this can take enough time that your users might think the system has a problem.

   If it isn't possible to have the form on its own HTML page, make sure the form is easily identifiable on the page. For example, give the form a name or use CGI data (the text that the follows the question mark in the URL) to identify the page and form.
2. In the Administration Console, click *Access Manager* > *Policies* > *New*.
3. Specify a name for the policy, select *Access Gateway: Form Fill* as its *Type*, then click *OK*.
4. In the *Actions* section, click *New* and select *Form Fill*.


Actions
New
Do Form Fill Form Selection
Form Name
:
CGI Matching Criteria... [No items]
Page Matching Criteria... [No items]
Fill Options
New
Input Field Name
Input Field Type
Input Field Value
Data Conversion
Text
Credential Profile
:
LDAP Credentials:LDAP User Name
[None]
Submit Options
☐ Auto Submit
☐ Debug Mode
☐ Mask Data
☐ Insert Text in Header
Text to Insert... [No items]
☐ Enable JavaScript Handling
Functions to Keep... [No items]
Statements to Execute on Submit... [No items]
Error Handling
Redirect to URL:
Changes made on this panel must be applied from the Policies Panel.
OK
Cancel

If you are converting an iChain® Form Fill policy written in XML to an Access Gateway policy, see "URLs Requiring Form Fill" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

5 In the *Form Selection* section, specify how the Access Gateway can identify the form on the page. Select one or more of the following methods. To be as specific as possible, use as few of these methods as possible. For information on how to use these options effectively, see "Creating a Form Matching Rule" on page 488.

Form Name: Identities the form on the HTML page. Select one of the following:

- **Form Name:** If the `<form>` element on your HTML page specifies a name attribute, select *Form Name* and specify the value of the name attribute in the text box. For example, suppose your form contains the following:

  ```
  <form name="mylogin" action="validatepassword.php"
  method="post" id="form1">
  ```

  For this form, you would specify *mylogin* in the text box.

- **Form Number:** The Access Gateway numbers forms sequentially from the top of the HTML page. If your page has multiple forms, you can use *Form Number* option and specify the form's sequential location in the text box.

- **Form ID:** If the `<form>` element on your HTML page specifies an id attribute, select *Form ID* and specify the value of the id attribute in the text box. For example, suppose your form contains the following

  ```
  <form name="mylogin" action="validatepassword.php"
  method="post" id="form1">
  ```

  For this form, you would specify *form1* in the text box.

CGI Matching Criteria: Allows the Access Gateway to evaluate the query string in the URL (the portion after the question mark) to differentiate pages that have the same URL. Consider the following URL:

`http://webaccess.novell.com/servlet/webacc?Action=User.login`

For this URL, enter the following string in the text box for *CGI Matching Criteria*:

`Action=User.login`

If possible, copy the text from the form and paste it into the *CGI Matching Criteria* text box.

Page Matching Criteria: Causes the Access Gateway to search the HTML page for the specified text. If the specified text is found on the page, the page is a match for the policy. If it isn't found, the page is not a match for the policy and the policy is not applied. For example, suppose your HTML page has the following string within the `<FORM>` element:

`<title>Form Fill Test Page</title>`

If you enter this string in the *Page Matching Criteria* box, the Access Gateway searches the form for this string. If it finds the string, it knows it has a match.

White space is significant. If the text in the text box is left-justified, the text can be found anywhere on the HTML page. If the text contains leading white space, such as ten spaces, the text must be found with ten leading spaces. If possible, copy the text as it appears on the form and paste it into *Page Matching Criteria* text box.

The more specific your information is, the faster Access Gateway can match the form. Parsing page matching criteria is a very intensive process. If possible, use the URL path specified for the protected resource or *CGI Matching Criteria* to identity the form.

6. In the *Fill Options* section, create an entry for all the input fields and select options in the form. For each input field or select option, you need to specify the following information:

   Input Field Name: Specifies the name of the field or option. This is the name attribute of the element on the form.

   Input Field Type: Specifies the type attribute for the input field or select option in the form. Select one of the following data types for the field:

   - **Text:** Indicates that the field is a text field on the form.
   - **Password:** Indicates that the field is a password field on the form.
   - **Checkbox:** Indicates that the field is a check box on the form.
   - **Radio Button:** Indicates that the field is a radio button on the form.
   - **Select:** Indicates that the field is a select option on the form.
   - **Hidden:** Indicates that the field is an input field, but that this field is hidden from the user.
   - **Not Specified:** Indicates that the field is an input field, but the data type is not specified in the form.

   Input Field Value: Specify the value for the field. You must specify the data type, then enter the value. Select one of the following data types:

   - **Credential Profile:** Specifies that the value should be retrieved from the credentials the user specified during authentication.
     - **LDAP Credentials:** If you prompt the user for a username and password, select this option, then either *LDAP User Name* (the cn of the user) or *LDAP User DN* (the fully distinguished name of the user). Your Web server requirements determine which one you use.

       The default contracts assign the cn attribute to the Credential Profile. If you create your own authentication contract, you can assign a different attribute to the Credential Profile.

    If your user store is an Active Directory server, you need to be aware that the cn attribute is used even though the user login is chosen from the SAMAccountName attribute. If you want to use the SAMAccountName attribute in the Credential Profile, you need to create your own authentication contract.

    - **X509 Credentials:** If you prompt the user for a certificate, select this option, then select one of the following. Your Web server requirements determine which one you use.

      X509 Public Certificate Subject: Specifies that the subject field from the certificate should be the value, which can match the DN of the user, depending upon who issued the certificate.

      X509 Public Certificate Issuer: Specifies that the issuer field from the certificate should be the value, which is the name of the certificate authority (CA) that issued the certificate.

      X509 Public Certificate: Specifies that the entire certificate should be the value.

      X509 Serial Number: Specifies that the certificate serial number should be the value.

    - **SAML Credential:** Injects the SAML assertion as the value of the field when SAML is used for authentication. This value is usually used for the user's password.

- **LDAP Attribute:** Indicates that the value should be retrieved from the specified LDAP attribute. If the attribute you require does not appear in the list, click *New* to add the attribute.
- **Liberty User Profile:** Indicates that the input field contains a Liberty User Profile attribute. In the value field, select the attribute. The attribute you select must be mapped to an LDAP attribute, and the Access Gateway retrieves its value from the LDAP directory.
- **Shared Secret:** Indicates that the input field contains a user-entered value that is to be stored in the specified shared secret store.

  You can create your own value. Click *New Shared Secret*, specify a display name for the store, and the Access Manager creates the store. Select the store, click *New Shared Secret Entry*, specify a name for the attribute, then click *OK*. The store can contain one name/value pair or a collection of name/value pairs. For more information, see Section 30.4, "Creating and Managing Shared Secrets," on page 499.

- **String Constant:** Indicates that the input field contains a static value. In the text box, enter the value for the string constant.

Data Conversion: Specify whether the case of the value entered by the user should be converted. Select one of the following options:

- **None:** Indicates that no conversion should be performed on the value.
- **To Upper Case:** Indicates that the value should be converted to uppercase.
- **To Lower Case:** Indicates that the value should be converted to lowercase.
- **LDAP DN to NDAP Partial Dot Notation:** Converts the LDAP DN (which uses typed, comma notation) to eDirectory™ typeless, dot notation: For example:

  `cn=jsmith,ou=Sales,o=novell to jsmith.sales.novell`

- **LDAP DN to NDAP Leading Partial Dot Notation:** Converts the LDAP DN to eDirectory typeless, leading-dot notation.

  `cn=jsmith,ou=Sales,o=novell to .jsmith.sales.novell`

- **LDAP DN to NDAP Fully Qualified Partial Dot Notation:** Converts the LDAP DN to eDirectory typed, dot-notation. For example:

  `cn=jsmith,ou=Sales,o=novell to cn=jsmith.ou=Sales.o=novell`

7. In the *Submit Options* section, specify how you want the information in the form submitted to the Web server. (The HTML form page determines whether the post method or the get method is used for the submission) Select one or more of the following options:

   Auto Submit: Indicates that you want the form submitted to the Web server without having the user confirm the submission by clicking a *Submit* button. If this option is not selected, Form Fill can fill in the data, but the user must click the *Submit* button before the data is sent to the Web server. If you select *Auto Submit*, you can select one or more of the following options:

   - **Debug Mode:** Allows you to verify that the information in the filled-in form is valid before it is posted to the Web server. You can right-click and view the source that is being submitted to the Web server. If it is correct, click *Submit* to send it to the Web server.

     This is a troubleshooting option. We recommend that you use it when creating a new Form Fill policy, and that you remove it when you have determined that the policy is behaving as expected.

   - **Mask Data:** Replaces text input field values (username, password, etc.) with nov-ss-ff-masked instead of the value specified by the value parameter when the form is sent to the browser. The Access Gateway replaces these masked values with the real values when the Access Gateway submits the form to the Web server. The user's browser never sees the actual values for these fields.

   Insert Text in Header: If this option is selected, you can use the *Text to Insert* option to specify text to add to the header. Use this option to insert static values into the form.

   Enable JavaScript Handling: Retains JavaScript from the original page. Use the following fields to specify how you want the JavaScript handled:

   - **Functions to Keep:** Specifies the functions you want executed from the JavaScript on the original page. In the text box, use the following format:

     ```
     function setCookie
     ```

     where `function` is a key word, followed by a space, and then the name of the function. Each function should be entered on a separate line, but you need only one function per script block. Everything must match exactly (name, capitalization, white space.) If possible, copy the function name from the HTML page.

   - **Statements to Execute on Submit:** Specifies the functions you want executed just before the form is posted. Copy the JavaScript from the HTML page into this text box or add a Java function that you want called that is not on the HTML page. This allows you to modify the behavior of the form when you can't modify the form.

     If the text box is empty, the JavaScript function specified in the submit field of the HTML page executes before the form is posted.

   For more information, see "Including JavaScript in a Form Fill Policy" on page 490.

8. In the *Error Handling* section, specify how you want errors handled.

   Redirect to URL: When an LDAP or NSS error occurs, the user is redirected to the URL you specify in the text box. This is optional and allows you to customize the error handling process. If you do not customize it, a standard error page is displayed.

9. Click *OK*, then click *Apply Changes*.

10. Continue with Section 13.4.4, "Assigning a Form Fill Policy to a Protected Resource," on page 213 or Section 30.3.3, "Creating a Login Failure Policy," on page 497.

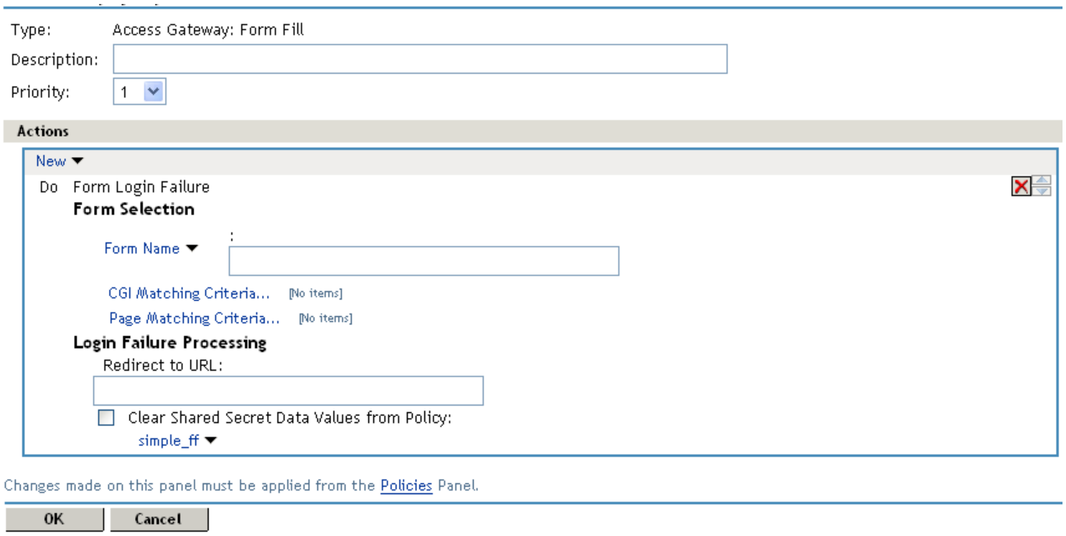
30.3.3 Creating a Login Failure Policy

The Login Failure policy can be part of the same policy as the Form Fill policy, if both share the same URL. In this case, the Form Login Failure policy should be the first action in the policy, and the Form Fill policy should be the second action in the policy. This causes a login failure to execute the policy that clears the stored data and the Form Fill policy to prompt the user for new data.

If the user is redirected to a different page when login fails, it is best to create a separate policy for that page, create a protected resource that includes just that page, and assign your Form Login Failure policy to that resource.

To create a Login Failure policy:

1. In the Administration Console, click *Access Manager* > *Policies* > *New*.
2. Specify a name for the policy, select *Access Gateway: Form Fill* as its *Type*, then click *OK*.
3. In the *Actions* section, click *New*, then select *Form Login Failure*.



Type: Access Gateway: Form Fill

Description:

Priority: 1

Actions

Do Form Login Failure

Form Selection

CGI Matching Criteria... [No items]

Page Matching Criteria... [No items]

Login Failure Processing

Redirect to URL:

☐ Clear Shared Secret Data Values from Policy: simple_ff

Changes made on this panel must be applied from the Policies Panel.

4. In the *Form Selection* section, identify the form. This section uses the same criteria for identifying a form as the Form Fill policy. For more information, see Step 5 on page 493 and "Creating a Form Matching Rule" on page 488.
5. In the *Login Failure Processing* section, define the actions you want executed when a user fails to log in. Fill in the following fields:

   Redirect to URL: When a user's login attempt fails, use this option with its text box to specify the URL you want the user redirected to. This is optional and allows you to customize what happens on login failures.

   Clear Shared Secret Data Values From Policy: Select this field to delete the user's stored data for a Form Fill policy. If the user has the ability (and perhaps the requirement) to periodically change his or her password or any other information on the form, you need to select this field. Otherwise, the wrong data can be stored for the user, and the Access Gateway has no way of updating the information.

   From the list of Form Fill policies, select the policy whose stored values should be cleared with this Login Failure policy.
6. Click *OK*, then click *Apply Changes*.

7 Continue with Section 13.4.4, “Assigning a Form Fill Policy to a Protected Resource,” on page 213.

30.3.4 Troubleshooting a Form Fill Policy

When a new Form Fill policy is not behaving as expected, use the following tips to discover the cause:

- Select the *Debug Mode* option. This option prepares the form for submission, but doesn’t submit the form until you click the *Submit* button. This allows you to view the source, and determine if the policy is generating the required data.
- Check to ensure that all input fields have valid names, that the fields are being filled in the correct order, and that any JavaScript commands have been entered correctly.
- Enable Form Fill logging. Form Fill is a function of both the proxy service and the embedded service provider. The embedded service provider logs the evaluation of the policy, and the proxy logs the process of gathering the data. To enable the embedded service provider tracing, see Section 39.1, “Turning on Logging for Policy Evaluation,” on page 599. To enable Access Gateway log entries for Form Fill policies, see “Enabling Form Fill Logging” on page 614.

Check for the following problems with the source content of the Form Fill page:

- “Valid HTML Structure” on page 498
- “The Option Element Does Not Contain a Value Attribute” on page 498
- “The Form Element Does Not Contain a Method Attribute” on page 499

Valid HTML Structure

The Form Fill process aborts if the page does not contain valid HTML structure. The page must contain the `<html></html>` tags, and the form must contain the `<form></form>` tags. If these tags are missing, you should correct the source page on the Web server. If this is not possible, you can create a rewriter policy to add the tags.

- To add the `<html>` tag, have the rewriter policy search for the `<body>` tag, and replace it with `<html><body>`.
- To add the `</html>` tag, have the rewriter policy search for the `</body>` tag, and replace it with `</body></html>`.
- Use similar entries to add the `<form></form>` tags. You’ll need to discover which tag or phrase starts and stops the form.

Configure your rewriter policy so that it runs before the default rewriter policy.

The Option Element Does Not Contain a Value Attribute

If an `<option>` element does not contain a value attribute, Form Fill cannot fill the value. For example:

```
<form action="select.htm">
   <select name="top2">
      <option>Bob</option>
      <option>Alice</option>
   </select>
</form>
```

If your form contains `<option>` elements similar to these, they need to rewritten to contain a value attribute. For example:

```
<form action="select.htm">
   <select name="top2">
      <option value="name1">Bob</option>
      <option value="name2">Alice</option>
   </select>
</form>
```

If possible, change the source page on the Web server to add the value attribute to the `<option>` elements. If this is not possible, you can use a rewriter policy to add the value attribute.

- For the Bob option, have the rewriter policy search for `<option>Bob` and replace it with `<option value="name1">Bob`.
- For the Alice option, have the rewriter policy search for `<option>Alice` and replace it with `<option value="name1">Alice.`

Configure your rewriter policy so that it runs before the default rewriter policy.

The Form Element Does Not Contain a Method Attribute

If the `<form>` element does not contain a method attribute, Form Fill does not run an Auto Post. For example, the following form cannot use an Auto Post.

```
<form name="loginForm">
```

To enable Form Fill so that it can run an Auto Post, you need to add a method attribute to the <form> element. For example:

```
<form method="get" action="index.htm" name="loginForm">
```

If possible, change the source page on the Web server to add the method attribute to the `<form>` element. If this is not possible, you can use a rewriter policy to add the method attribute.

- Search for `<form`
- Replace this string with `<form method="get" action="index.htm"`

Configure your rewriter policy so that it runs before the default rewriter policy.

30.4 Creating and Managing Shared Secrets

A shared secret is an object that holds name and value pairs for Form Fill and Identity Injection policies.

- If your HTML form prompts the user for more than credential information, you need to create a shared secret to store the values.
- If your Web server requires some name/value pairs to be injected and these are not available from the HTTP request, you need to create a shared secret to store these name/value pairs so that they can be injected into the header before it is sent to the Web server.

Access Manager supports the creation and use of local secret stores and the use of remote secret stores if you are using eDirectory as your user store and have set up a secret store there. The local secret store can only be used with Access Manager. If you want users to have access to the secret store without authenticating to Access Manager, you must configure the user store to use Novell

SecretStore. For more information on configuring the user store to use secret store, see Section 8.1.4, “Configuring a User Store for Secrets,” on page 94.

This section describes

- Section 30.4.1, “Naming Conventions for Shared Secrets,” on page 500
- Section 30.4.2, “Creating a Shared Secret Independent of a Policy,” on page 500
- Section 30.4.3, “Modifying and Deleting a Shared Secret,” on page 501

30.4.1 Naming Conventions for Shared Secrets

The policy engine allows you to create shared secrets and name the attributes for the store as you are creating an Identity Injection or Form Fill policy. When creating the shared secret, we recommend that you name the shared secret after the application for which you are creating the policy. Each value requires a name, and we recommend that you use the same name for the value name as needed for the *Input Field Name* on a Form Fill policy or for the header name on an Identity Injection policy. For example if your e-mail application requires the e-mail address for the name on the login form, you could set up the following Shared Secret values:

Input Field Name	Input Field Value	Shared Secret Name	Entry Name
emailaddress	Shared Secret	emailapp	emailaddress

Your applications, how you use them, and your personal preferences determine whether you create one shared secret and use it for all your applications or whether you create a shared secret for each application.

- If the applications use some of the same secrets, you can use the same shared secret for these applications. In this case, give the shared secret a name that reflects all of the applications using it.
- If an application does not use the same secrets as another application and you want the freedom to remove the application and its secrets without affecting other applications, you should create a separate shared secret for this application.
- If you are using Novell® SecretStore®, then secret names specified in your Access Manager policies need to match the names you have already configured.

A local shared secret store does not contain any name/value pairs until you configure a Form Fill policy to add name/value pairs or enable the *Allow End Users to See Credential Profile* option. This option allows the username and password to be stored in the local secret store.

30.4.2 Creating a Shared Secret Independent of a Policy

You can create a shared secret as part of the process of creating a Form Fill or Identity Injection policy. You can also create a shared secret independent of a policy:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Shared Settings* > *Custom Attributes*.
2. To create a new shared secret, click *New* in the *Shared Secret Names* section, and fill in the following fields:

   Secret Name: Specify a display name for the shared secret.

Secret Entry Name. Specify an attribute name for a value you want to store.

3. Click *OK*.

   The Identity Server creates and encrypts the object.
4. To create additional attributes to store values, repeat Step 2 and Step 3.
5. Click *OK*.

30.4.3 Modifying and Deleting a Shared Secret

Before deleting a shared secret, you need to delete the policies that are using the shared secret or modify the policies to use a different shared secret. For information about deleting policies, see Section 26.4.2, “Deleting Policies,” on page 387.

Both Form Fill and Identity Injection policies can use shared secrets. The following instructions explain how to modify an Identity Injection policy to use a new shared secret and then how to delete the old shared secret.

1. In the Administration Console, click *Access Manager* > *Policies* > *[Name of Policy]* > *[Rule]*.
2. Select the *Value* field that uses the shared secret you want to delete. Click its name, then *New Shared Secret*.
3. Specify the name for a new shared secret, then click *OK*.
4. Click the name of the shared secret, select the new shared secret store, then *New Shared Secret Entry*.
5. Specify the attribute name for this shared secret entry, then click *OK*.
6. Modify any other Value fields to use the new shared secret. Create new attributes as needed.
7. To save the modifications to the policy, click *OK* twice, then *Apply Changes*.
8. To delete the old shared secret, click *Identity Servers* > *Shared Settings* > *Custom Attributes*.
9. Select the name of the shared secret and the attributes, then click *Delete*.

30.5 Importing and Exporting Form Fill Policies

You can import and export Form Fill policies in order to run them in other Access Manager configurations and to analyze the policy. The policy is exported as a text file with XML tags. We do not recommend editing the exported file with a text editor. Any changes you want to make to a policy ought to be done through the Administration Console.

To export a Form Fill policy:

1. In the Administration Console, click *Access Manager* > *Policies*.
2. Select a Form Fill policy, then click *Export*.
3. (Optional) Modify the name suggested for the file.
4. Click *OK*.
5. Using the features of your browser, specify where the file is be copied.

To import a policy:

1. Make sure any referenced shared secret stores have been created. See Section 30.4, “Creating and Managing Shared Secrets,” on page 499.

2. If the policy uses LDAP or Liberty Profile attributes, make sure the Identity Server has been configured for these same attributes.
3. In the Administration Console, click *Access Manager* > *Policies*.
4. If the policy uses LDAP or Liberty Profile attributes, make sure the Identity Server has been configured for these same attributes.
5. Click *Import*, then browse to the location of the file.
6. Click *OK*.
7. When the policy appears in the list, click *Apply Changes*.

Monitoring Access Manager Components

VII

This section describes the various ways you can determine whether the Access Manager is functioning normally and whether an Internet attack is in progress. This section discusses the following topics:

Enabling Auditing

31

Access Manager includes a licensed version of Novell® Audit to provide compliance assurance logging and to maintain audit log entries that can be subsequently included in reports. In addition to selectable events, device generated alerts are automatically sent to the audit server.

Audit logs record events that have occurred in the identity and access management system and are primarily intended for auditing and compliance purposes. The types of events that are logged include the following:

- Starting, stopping, and configuring a component
- Success or failure of user authentication
- Role assignment
- Allowed or denied access to a protected resource
- Error events
- Denial of service attacks
- Security violations and other events necessary for verifying the correct and expected operation of the identity and access management system.

Audit logging does not track the operational processing of the Access Manager components; that is, the processing and interactions between the Access Manager components required to fulfill a user request. (For this type of logging, see Section 32.2, "Configuring Identity Server Logging," on page 516). Audit logs record the results of user and administrator requests and other system events. Although the primary purpose for audit logging is for auditing and compliance, the types of events logged can also be useful for detecting abnormal and error conditions and can be used as a first alert mechanism for system support. You can configure the audit log entries to generate alerts by leveraging the Novell Audit Notification feature. You can select to generate e-mail, syslog, and SNMP notifications.

Access Manager has been assigned the Novell Audit server-alert event code 0x002E0605. The Novell Audit Platform Agent is responsible for packaging and forwarding the audit log entries to the configured Novell Audit server. If the Novell Audit server is not available, the platform agent caches log entries until the server is operational and can accept audit log data.

For additional information about Novell Audit, see Novell Audit 2.0.2 (http://www.novell.com/documentation/novellaudit20/index.html) at the Novell Documentation Web site.

This section describes the following Access Manager features of auditing:

- Section 31.1, "Configuring Access Manager for Novell Auditing," on page 506
- Section 31.2, "Enabling Identity Server Audit Events," on page 508
- Section 31.3, "Enabling Access Gateway Audit Events," on page 510
- Section 31.4, "Enabling SSL VPN Audit Events," on page 511
- Section 31.5, "Querying Data and Generating Reports in Novell Audit," on page 512

For a listing of all Novell Audit events logged by Access Manager, see Appendix G, "Access Manager Audit Events and Data," on page 721.

31.1 Configuring Access Manager for Novell Auditing

By default, Access Manager is preconfigured to use the Novell Audit server it installs on the first instance of the Administration Console. If you install more than one instance of the Administration Console for failover, Novell Audit is installed with each instance. However, if you already use Novell Audit, you can continue using your existing installation with Access Manager. You'll need to configure Access Manager to use your audit server. You'll also need to register the Access Manager with your audit servers by importing the `nids_en.lsc` and `sslvpn_en.lsc` files.

Novell Access Manager allows you to specify only one Novell Audit server. You still have failover, when the audit server goes down. The auditing clients on the Novell Access Manager components go into caching mode when the audit server is not available. They save all events until the entries can be sent to the audit server.

This section includes the following topics:

- Section 31.1.1, "Specifying the Logging Server and Events," on page 506
- Section 31.1.2, "Configuring the Platform Agent," on page 507
- Section 31.1.3, "Generating Queries," on page 507

31.1.1 Specifying the Logging Server and Events

The Secure Logging Server manages the flow of information to and from the Novell auditing system. It receives incoming events and requests from the Platform Agents, logs information to the data store, monitors designated events, and provides filtering and notification services. It can also be configured to automatically reset critical system attributes according to a specified policy.

1 To specify the logging server, click *Access Manager* > *Auditing*.

2 Fill in the following fields:

   Server: Specify the IP address or DNS name of the audit logging server you want to use. By default, the system uses the primary Administration Console IP address. If you want to use a different Secure Logging Server, specify that server here.

   Access Manager does not currently support the use of custom application certificates. (For information on this Novell Audit feature, see Authentication Logging Applications (http://www.novell.com/documentation/nsureaudit/nsureaudit/data/am8ewv2.html).)

   To use Novell Sentinel™ instead of Novell Audit, specify the IP address or DNS name of your Collector. For more information on Sentinel, see Sentinel 6 (http://www.novell.com/documentation/sentinel6/index.html).

   Port: Specify the port that the Platform Agents use to connect to the Secure Logging Server.

   To use Novell Sentinel instead of Novell Audit, specify the port of your Collector.

   IMPORTANT: Whenever you change the port or address of the Secure Logging Server, all Access Gateways must be updated, then every Access Manager device (Identity Server, Administration Console, Access Gateways, SSL VPN servers, and J2EE Agents) must be rebooted (not just the module stopped and started) before the configuration change takes affect.

3 Under *Management Console Audit Events*, specify the system-wide events you want to audit:

   Select All: Selects all of the audit events.

Health Changes: Generated whenever the health of a server changes.

Server Imports: Generated whenever a server is imported into the Administration Console.

Server Deletes: Generated whenever a server is deleted from the Administration Console.

Configuration Changes: Generated whenever you change a server configuration.

4. Click *OK*.

   If you did not change the address or port of the Secure Logging Server, this completes the process. It may take up to fifteen minutes for the events you selected to start appearing in the audit files.

   If you changed the address or the port of the Secure Logging Server, complete the following steps:

5. If the Administration Console is the only Access Manager component installed on the machine, edit the Novell Audit Configuration file.

   For security reasons, this file cannot be edited from the Administration Console when it is the only Access Manager component on the machine.

   Edit the `/etc/logevent.conf` file and specify the new address and port of the Secure Logging Server.

6. Restart the Administration Console. From a terminal window, enter the following command:

   ```
   /etc/init.d/novell-tomcat4 restart
   ```

7. Restart every device imported into the Administration Console.

   The devices (Identity Server, Access Gateway, SSL VPN, J2EE Agents) do not start reporting events until they have been restarted.

31.1.2 Configuring the Platform Agent

The Platform Agents installed with the Access Manager components use an embedded certificate. Access Manager does not currently support the use of custom application certificates. For information on this Novell Audit feature, see Authenticating Logging Applications (http://www.novell.com/documentation/nsureaudit/nsureaudit/data/am8ewv2.html).

The platform agents that are installed on each Access Manager component can be configured by modifying the `logevent` file. For the location of this file and its parameters, see Logevent (http://www.novell.com/documentation/nsureaudit/nsureaudit/data/al36zjk.html#alibmyw).

IMPORTANT: Do not use this file to modify the IP address of the Secure Audit Server. Use the Administration Console for this task (see Section 31.1.1, “Specifying the Logging Server and Events,” on page 506).

If you are using Sentinel, do not modify this file. The parameters in this file should be set on the collector.

31.1.3 Generating Queries

Queries let you create, run, edit and delete queries and event verifications. You can create two kinds of queries in Access Manager: manual queries and saved queries. Manual queries are simply queries that are not saved; they only run one time. All verification queries are saved. Saved queries and verifications are listed in the Queries list and can be run again and again against different databases.

Access Manager uses queries to request information from MySQL* and Oracle* databases. All queries are defined in SQL. Although you must be familiar with the SQL language to create SQL query statements, this is the most powerful and flexible query method.

For information about queries, see Novell Audit 2.0.2 (http://www.novell.com/documentation/novellaudit20/index.html), at the Novell Documentation Web site.

31.2 Enabling Identity Server Audit Events

All user and administrator actions can be logged to Novell Audit. You can generate a Novell Audit logging event to indicate whether authentications are successful or unsuccessful. The following steps assume that you have already set up Novell Audit on your network. For more information, see Section 31.1, "Configuring Access Manager for Novell Auditing," on page 506

1. In the Administration Console, click *Access Manager > Identity Server > Servers > Edit > Logging*.
2. In the *Novell Audit Logging* section, select *Enabled*.
3. Select the events for notification.

   Select All: Select this option for all events. Otherwise, select one or more of the following:

Event	Description
Login Provided	Generated when an identity provider sends authentication to a service provider. Role assignment audit events are included in authentication audit events for the identity server.
Login Provided Failure	Generated when an identity provider attempts to send authentication to a service provider but fails.
Login Consumed	Generated when the Identity Server is authenticated either locally or by an external identity provider. Role assignment audit events are included in authentication audit events for the identity server.
Login Consumed Failure	Generated when the Identity Server initiates authentication, but the process fails.
Logout Provided	Generated when an identity provider sends a logout request to a service provider that it has authenticated.
Logout Local	Generated when the Identity Server receives a command to log out from the user.
Federation Request Sent	Generated when a service provider attempts to federate with an identity provider.
Federation Request Handled	Generated by the Identity Server when processing a request for federation.

Event	Description
Defederation Request Sent	Generated by the identity provider when a request for defederation is sent to another provider.
Defederation Request Handled	Generated when the Identity Server processes a request for defederation.
Register Name Request Handled	Generated when the Identity Server processes a request for changing a name identifier.
Attribute Query Request Handled	Generated by the Identity Server when processing an attribute request from a service provider.
Web Service Query Handled	Causes a Web service query request to be sent to an identity provider.
Web Service Modify Handled	Causes a Web service modify request to be sent to an identity provider.
User Account Provisioned	Generated by the Identity Server when functioning as an identity consumer and when an account has been provisioned.
User Account Provisioned Failure	Generated by the Identity Server when functioning as an identity consumer and when account provisioning has failed.
Ldap Connection Lost	Generated when the LDAP connection is lost.
Ldap Connection Reestablished	Generated when the LDAP connection is reestablished.
Server Started	Generated when the server gets a start command from the server communications module.
Server Stopped	Generated when the server gets a stop command from the server communications module.
Server Refreshed	Generated when the server gets a refresh command from the server communications module.
Intruder Lockout Detected	Generated when an attempt to log in as a particular user with an invalid password has occurred more times than is allowed by the directory.
Component Log Severe Messages	Logged for all component messages with level of Severe.
Component Log Warning Messages	Logged for all component messages with level of Warning.

4 Click *Apply*, then *OK*.

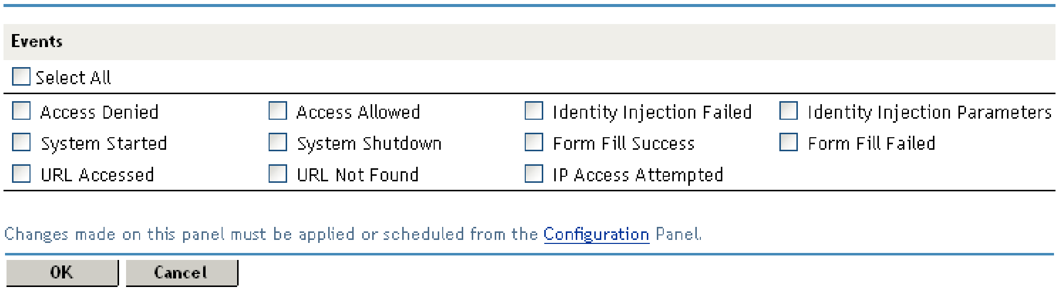
5 Click *Servers* > *Update Servers*.

Restart the Novell Audit server.

31.3 Enabling Access Gateway Audit Events

The *Novell Audit* option in the Access Gateway allows you to configure the events you want audited. The following steps assume that you have already set up Novell Audit on your network. For more information, see Section 31.1, “Configuring Access Manager for Novell Auditing,” on page 506.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Novell Audit*.



Events

☐ Select All

☐ Access Denied
☐ System Started
☐ URL Accessed
☐ Access Allowed
☐ System Shutdown
☐ URL Not Found
☐ Identity Injection Failed
☐ Form Fill Success
☐ IP Access Attempted
☐ Identity Injection Parameters
☐ Form Fill Failed

Changes made on this panel must be applied or scheduled from the Configuration Panel.

2 Select the events for notification.

Select All: Select this option for all events. Otherwise, select one or more of the following:

Event	Description
Access Denied	Generated when a requested action is denied because the requester has insufficient access rights to a URL.
System Started	Generated when the Access Gateway is started.
URL Accessed	Generated when a user accesses a URL.
Access Allowed	Generated when a requested action is allowed because the requester has the correct access rights to a URL.
System Shutdown	Generated when the Access Gateway is stopped.
URL Not Found	Generated when a requested URL cannot be found.
Identity Injection Failed	Generated when an Identity Injection policy fails to obtain a requested value to inject into the HTTP header.
Form Fill Success	Generated when a Form Fill policy successfully fills in a form.
IP Access Attempted	Generated when a user attempts to access a URL with an IP address instead of the published DNS name configured in the Access Gateway.
Identity Injection Parameters	Generated when the Identity Injection policy successfully injects data into the HTTP header. Some of the data might be injected with the value field empty. When this happens, this event should also produce an *Identity Injection Failed* event.
Form Fill Failed	Generated when a Form Fill policy fails to successfully fill in a form.

3 To save your modifications, click *OK* twice.

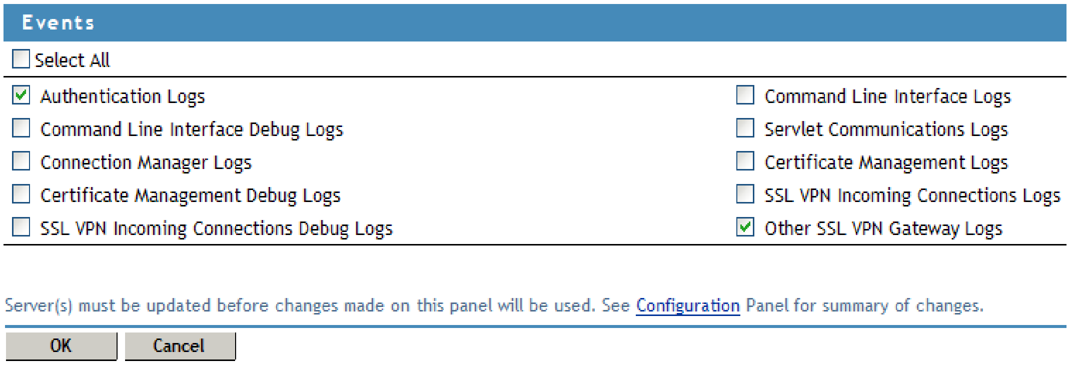
4 On the Access Gateways page, click *Update*.

31.4 Enabling SSL VPN Audit Events

The *Novell Audit Settings* option allows you to configure the events you want audited. The following steps assume that you have already set up Novell Audit on your network. For more information, see Section 31.1, “Configuring Access Manager for Novell Auditing,” on page 506.

1 In the Administration Console, click *Access Manager* > *SSL VPNs* > *Edit*.

2 Select *Novell Audit Settings* from the *Novell Audit and Alerts* section. The Novell Audit Settings for SSL VPN page is displayed.



☐ Select All

☑ Authentication Logs
☐ Command Line Interface Debug Logs
☐ Connection Manager Logs
☐ Certificate Management Debug Logs
☐ SSL VPN Incoming Connections Debug Logs
☐ Command Line Interface Logs
☐ Servlet Communications Logs
☐ Certificate Management Logs
☐ SSL VPN Incoming Connections Logs
☑ Other SSL VPN Gateway Logs

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

3 Select the *Select All* option to receive logs for all the events. Otherwise, select one or more of the following:

Event	Description
Authentication Logs	Generates a log file with the authentication details.
Command Line Interface Logs	Generates a log file with command line actions.
Command Line Interface Debug Logs	Generates a log file with command line actions.
Servlet Communications Logs	Generates a log file with information on servlet communication.
Connection Manager Logs	Generates a log file containing information on the connection activity.
Certificate Management Logs	Generates a log file with certificate management information.
Certificate Management Debug Logs	Generates a log file with certificate management information.
SSL VPN Incoming Connections Logs	Generates a log file containing information on the incoming connection.
SSL VPN Incoming Connections Debug Logs	Generates a log file containing debug information on the incoming connection.
Other SSL VPN Gateway Logs	Generates a log file containing miscellaneous information.

4 To save your modifications, click *OK,* then click *Apply Changes* on the Configuration page.

31.5 Querying Data and Generating Reports in Novell Audit

Novell Audit provides two tools to query events and generate reports: the Novell Audit iManager plug-in and Novell Audit Report (`LReport`).

The following sections provide more information on these tools:

- Section 31.5.1, "The Novell Audit iManager Plug-in," on page 512
- Section 31.5.2, "Novell Audit Report," on page 512

31.5.1 The Novell Audit iManager Plug-in

The Novell Audit iManager plug-in is a Web-based JDBC* application that enables you to query MySQL* and Oracle* databases. All queries are defined in SQL.

iManager includes several predefined queries and it includes a Query Builder to help you define basic query statements. Of course, you can also build your own SQL query statements.

For basic steps in configuring the Auditing and Logging plug-in on the Administration Console, see "Creating Novell Audit Queries" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

For complete information on defining and running queries in iManager, see the following sections in the Novell Audit 2.0 Administration Guide (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/bookinfo.html).

- Defining Your Query Databases in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#alost1z)
- Defining Queries in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#alpvc0a)
- Running Queries in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#alpv7ft)
- Verifying Event Authenticity in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#b34tzvi)
- Exporting Query Results in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#alqvrze)
- Printing Query Results in iManager (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpq2.html#alqvzva)

31.5.2 Novell Audit Report

Novell Audit Report is a Windows-based, ODBC-compliant application that can use SQL query statements or Crystal Decisions* Reports to query Oracle and MySQL data stores (or any other database that has ODBC driver support). You can define your own SQL query statements or import existing query statements and reports. Query results are returned in simple data tables; rows represent individual records and columns represent fields within those records.

For complete information on defining and running queries in Novell Audit Report, see the following sections in the Novell Audit 2.0 Administration Guide (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/bookinfo.html).

- Novell Audit Report Interface (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpgw.html#als9vcm)
- Defining Your Databases in Novell Audit Report (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpgw.html#als94w4)
- Verifying Event Authenticity in Novell Audit Report (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpgw.html#am9dbll)
- Working with Reports in Novell Audit Report (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpgw.html#alsn2fj)
- Working with Queries in Novell Audit Report (http://www.novell.com/documentation/novellaudit20/novellaudit20/data/alorpgw.html#alshpuw)

Configuring Logging

32

32.1 Understanding the Types of Logging

Access Manager supports three types of logging:

32.1.1 Component Logging for Troubleshooting Configuration or Network Problems

Each Access Manager component maintains log files that contain entries documenting the operation of the component. Component file logging records the processing and interactions between the Access Manager components that occur while satisfying user and administrative requests and during general system processing. By enabling the correct levels of logging for the various Access Manager components, an administrator can monitor how the Access Manager processes user and administrative requests. Transaction flows have been defined to help the administrator identify the processing steps that occur during the execution of specific types of user or administrative requests. All component file logs include tags and values that allow the administrator to identify and correlate which component file log entries pertain to a given transaction and user.

Component file logs are not primarily intended for debugging the software itself, although they can be used to detect software that is not behaving properly. Rather, the intent of component file logging is to document the operational processing of the Access Manager components so that system administrators and support personnel can identify and isolate problems caused by configuration errors, invalid user data, or network problems such as broken connection. However, component file logging is typically the first step in identifying software bugs.

Component file logging is more verbose than audit logging. It increases processing load, and on a day-to-day basis, it should be enabled only to log error conditions and system warnings. If a specific problem occurs, component file logging can be set to *info* or *config* to gather the information needed to isolate and repair the detected problem. When the problem is resolved, component file logging should be reconfigured to log only error conditions and system warnings.

Log files can be configured to include entries for the following events:

- Initialization and shutdown
- Configuration

- Events processed by the component, such as authentication, role assignment, resource access, and policy evaluation
- Error conditions

See Section 32.2, “Configuring Identity Server Logging,” on page 516.

32.1.2 Debug Trace Logging to Discover Software Problems

Debug trace logging is used to debug the software execution flow of an Access Manager component. Debug trace logging is the most verbose of the Access Manager logging categories and by its nature includes data that generally encompasses the information provided by both audit logging and component file logging. The information contained in debug trace logs can generally only be interpreted by those with access to the source code, such as Novell® support personnel or software engineers. System administrators might be required to enable debug trace logging in order to provide support personnel with the information necessary to resolve a software bug. Debug trace logging should not be enabled during normal operation of Access Manager.

See Section 32.3, “Configuring Debug Trace Logging,” on page 522.

32.1.3 HTTP Transaction Logging for Proxy Services

The Access Gateway allows you to log HTTP transactions. You can log what happens with an HTTP request and response during certain times:

- Between the browser and the Access Gateway
- Between the Access Gateway and the back-end Web server

You select fields from the HTTP header of a request and these fields are logged. You can then use these logged transactions to bill customers for Web services or to troubleshoot whether a request is refused because the browser didn’t send the required information or because the Access Gateway didn’t send the Web server the required information. This type of logging conforms to the W3C specification for proxy server logging in the common and extended log formats. This type of logging provides no information about the exchanges between the Access Gateway and the Identity Server. If you need to discover whether the Access Gateway is obtaining the correct information from the Identity Server for an Identity Injection or Form Fill policy, you need to turn on Component logging. See Section 32.2, “Configuring Identity Server Logging,” on page 516.

For HTTP transaction logging, see Section 32.4, “Configuring Access Gateway Logging,” on page 523.

32.2 Configuring Identity Server Logging

You can enable and configure how the system performs logging. Logging is the main tool you use for debugging the Identity Server configuration. All administrative and end-user actions and events are logged to a central event log. This allows easy access to this information for security and operational purposes. Additionally, the log system provides the ability to monitor ongoing activities (such as identity provider authentication activity, up-time of the system, and so on) using this page. File logging is not enabled by default.

Identity Servers, Access Gateways (Linux and NetWare®), and embedded service providers use these logging features. If you change or enable logging, you must update the Identity Server configuration (using Update Servers on the Servers page) and restart the service providers on the

Access Gateways, in order to apply the changes. When you disable logging, you must also restart the Access Gateway embedded service provider. See Section 3.2.7, "Rebooting the Access Gateway," on page 40.

This section describes the following about component logging:

- Section 32.2.1, "Enabling Component Logging," on page 517
- Section 32.2.2, "Downloading the Log Files," on page 518
- Section 32.2.3, "Managing Log File Size," on page 521

32.2.1 Enabling Component Logging

File logging records the actions that have occurred. For example, Web servers maintain log files listing every request made to the server. With log file analysis tools, it's possible to get a good idea of where visitors are coming from, how often they return, and how they navigate through a site. The content logged to file logging can be controlled by specifying logger levels and by enabling statistics logging.

1 In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Logging*.

2 The following options are available for component logging in the *File Logging* section:

   - **Enabled:** Enables file logging for this server and its associated embedded service providers.
   - **Echo To Console:** Copies the Identity Server log file to `/var/opt/novell/tomcat4/logs/catalina.out`. You can download the file from *Access Manager* > *Auditing* > *General Logging.* If you want to view Identity Server logs mixed with logs from other application devices, you use `catalina.out`.

     For the embedded service providers, it depends upon the platform:

     - For a Linux Access Gateway, this sends the messages to the `catalina.out` file of the Access Gateway.
     - For a NetWare Access Gateway, this sends the messages to the NetWare console.
     - For a SSL VPN, this sends the messages to the `catalina.out` file of the SSL VPN.
   - **Log File Path:** Specifies the path that the system uses to save the Identity Server XML log file. The default path is *`tomcat application directory`*`/web-inf/logs`.

     If you change this path, you must ensure that the user associated with configuring the identity or service provider has administrative rights to the Tomcat application directory in the new path.

     If you have a mixed platform environment (for example, the Identity Server is installed on Linux and the Access Gateway is on NetWare), do not specify a path. In a mixed platform environment, you must use the default path.
   - **Maximum Log Files:** Specifies the maximum number of log files to leave on the machine. After this value is reached, the system deletes log files, beginning with the oldest file. You can specify *Unlimited*, or values of 1 through 200. 10 is the default value.
   - **File Wrap:** Specifies the frequency (hour, day week, month) for the system to use when closing a log file and creating a new one. The system saves each file based on the time you specify and attaches the date and/or time to the filename.

- **GZip Wrapped Log Files:** Uses the GZip compression utility to compress logged files. The log files that are associated with the *GZip* option and the *Maximum Log Files* value are stored in the directory you specify in the *Log File Path* field.

3 In the Component File Logger Levels, you can specify the logging sensitivity for the following:

   Application: Logs system-wide events, except events that belong to a specific subsystem.

   Liberty: Logs events specific to the Liberty IDFF protocol and profiles.

   SAML 1: Logs events specific to the SAML1 protocol and profiles.

   SAML 2: Logs events specific to the SAML2 protocol and profiles.

   Web Service Provider: (Liberty) Logs events specific to fulfilling Web service requests from other Web service consumers.

   Web Service Consumer: (Liberty) Logs all events specific to requesting Web services from a Web service provider.

   Use the drop-down menu to categorize logging sensitivity. Higher logging levels include the lower levels in the log.

   - **Off:** Turns off component file logging for the selected item.
   - **Severe:** Logs serious failures that can cause system processing to not proceed.
   - **Warning:** Logs potential failures, but the impact on execution is minimal. Warnings indicate that you should be aware that this event is happening and might want to make a configuration change to avoid it.
   - **Info:** Logs informational events. No execution or data impact occurred.
   - **Verbose:** Logs static configuration information. The system logs any configuration errors under one of the primary three levels: Severe, Warning, and Info.
   - **Debug:** Includes all of the preceding levels.

4 (Optional) Enable statistics logging.

   When statistics logging is enabled, the system periodically sends the system statistics, in string format, to the current file logger. Statistical data (such as counts, levels, and so on) are included in the file log.

   4a In the *Statistics Logging* section, select *Enabled*.

   4b In the *Log Interval* field, specify the time interval in seconds that statistics are logged.

5 Click *OK*.

6 Update the Identity Server configuration (using *Update Servers* on the *Servers* page).

7 Restart the embedded service providers on the Access Gateways, in order to apply the changes.

   When you disable component logging, you need to update the Identity Server configuration and restart the embedded service provides.

32.2.2 Downloading the Log Files

The *General Logging* page displays the location of the files that the Access Manager components use for logging system messages. There are two exceptions:

- **J2EE Agent:** The J2EE Agent uses the J2EE global logger, and the location of this file is customizable. For information about J2EE agent log files, see "Viewing Log Files" in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*.

- **Default Auditing File:** If you have configured Novell Audit to send events to the default audit file (`/var/opt/novell/naudit/logs/auditlog`), this file does not appear in the list. If you want this file to appear in this list, make this file readable by the novlwww user.

  It is a breach of Novell Audit security for Access Manager code to change the permissions on this file. You must decide whether changing its permissions and displaying the file in this list compromises your security.

  To have it appear in the list of files for the Administration Console, configure the following:

  - Use commands similar to the following to grant the `novlwww` user executable permissions to the naudit directories:

    ```
    chmod o+x /var/opt/novell/naudit
    chmod o+x /var/opt/novell/naudit/logs
    ```

  - Use a command similar to the following to grant the `novlwww` user read access to the `auditlog` file:

    ```
    chmod o+r /var/opt/novell/naudit/logs/auditlog
    ```

To view or download the log file:

1. In the Administration Console, click *Auditing* > *General Logging*.
2. Click the link for the log file name, then either open it or save it to disk.

   You can use any text editor to view the file.

Each Access Manager Component generates multiple log files. Table 32-1 lists these files and the types of messages they contain.

Table 32-1 *Access Manager Log Files*

Component	Filename	Description
Identity Server		
	`/var/opt/novell/tomcat4/logs/catalina.out`	Logging to this file only occurs if you have selected the *Echo to Console* option from the *Identity Servers* > *Servers* > *Edit* > *Logging* page. When component logging has been set to info for Applications, it contains entries tracing user authentication and role assignment.
	`/opt/novell/devman/jcc/logs/jcc-0.log.0`	Contains the log entries for the server communications module related to interaction of the Identity Server with the Administration Console, such as imports, certificates, and configuration.
Administration Console		
	`/var/opt/novell/tomcat4/logs/catalina.out`	Contains Tomcat errors.

Component	Filename	Description
	`/opt/novell/devman/share/logs/app_sc.0.log`	Contains events related to importing devices, device configuration changes, health status changes, statistics reporting, and communication problems.
	`/opt/novell/devman/share/logs/app_cc.0.log`	Contains events related to policy configuration.
	`/opt/novell/devman/share/logs/platform.0.log`	Contains XML events for configuration changes. This log file contains very little useful information for system administrators.
Linux Access Gateway		
	`/var/log/novell/reverse/`<name>	If logging is enabled on one or more reverse proxies (see Section 32.4, "Configuring Access Gateway Logging," on page 523), this directory contains the log files. A directory is listed for each reverse proxy on which you have enabled logging.
	`/var/log/ics_dyn.log`	Contains all log entries generated by the Linux Access Gateway. Use syslog to control file rolling and log file distribution.
	`/opt/novell/devman/jcc/logs/jcc-0.log.0`	Contains the log entries for the server communications module related to interaction of the Access Gateway with the Administration Console, such as imports, certificates, and configuration.
	`/var/opt/novell/tomcat4/logs/catalina.out`	Logging to this file only occurs if you have selected the *Echo to Console* option from the *Identity Servers* > *Servers* > *Edit* > *Logging* page. Check this file for entries tracing the evaluation of authorization, identity injection, and form fill policies.
	`/var/log/lagsoapmessages`	Logs all the SOAP messages between the Linux Access Gateway and the embedded service provider.
	`/var/log/laghttpheaders`	Contains a log of the HTTP headers to and from the Linux Access Gateway.

Component	Filename	Description
NetWare Access Gateway		
	`log:\etc\proxy\data\logs\reverse\common\`	If logging is enabled on one or more reverse proxies (see Section 32.4, "Configuring Access Gateway Logging," on page 523), this directory contains the log files. A directory is listed for each reverse proxy on which you have enabled logging.
	`SYS:\etc\proxy\data\debug.log`	Contains the abend messages.
	`SYS:\jcc\logs\jcc-0.log.0`	Contains the log entries for the server communications module.
SSL VPN		
	`/var/opt/novell/tomcat4/logs/catalina.out`	Logging to this file only occurs if you have selected the *Echo to Console* option from the *Identity Servers* > *Servers* > *Edit* > *Logging* page.
	`/opt/novell/devman/jcc/logs/jcc-0.log.0`	Contains the log entries for the server communications module related to interaction of the SSL VPN with the Administration Console, such as imports, certificates, and configuration.
	`/var/log/messages`	Contains log entries for the connection manager and SOCKS server.
	`/var/log/novell-openvpn.log`	Contains log entries for the Enterprise mode tunneling components.
	`/var/log/stunnel.log`	Contains log entries for the Kiosk mode tunneling components.

For more information about the entries in the log files, see

- "Using the Log Files for Troubleshooting" on page 669
- "Understanding Policy Evaluation Traces" on page 600

32.2.3 Managing Log File Size

The logrotate daemon manages the log files located in the following directories:

`/var/opt/novell/tomcat4/logs`
`/opt/novell/roma/logs/`

The logrotate daemon has been configured to scan the files in these directories once a day. It rolls them over when they have reached their maximum size and deletes the oldest version when the maximum number of copies have been created.

If you want to modify this behavior, see the following files in the `/etc/logrotate.d` directory:

```
novell-tomcat4
novell-devman
```

For information about the parameters in these files, see the documentation for the logrotate daemon.

32.3 Configuring Debug Trace Logging

Novell recommends that you use the tracing feature only for software debugging. Sensitivity levels do not apply to trace logging. Therefore, you would not activate this feature during production, because it impacts processing speed. This feature is filterable by Java class or package.

To enable debug trace logging:

1. In the Administration Console, click *Access Manager* > *Identity Server* > *Servers* > *Edit* > *Logging*.
2. In the *File Logging* section, select *Enabled*.

   It is assumed that you have set up the Echo To Console, Log File Path, and File Wrap options when you set up component file logging. If you need help with these options, see Step 2 in Section 32.2.1, "Enabling Component Logging," on page 517.
3. In the *Trace Logging* section, select *Enabled*.

   This option enables trace logging and the *Custom Content Filter* link.
4. (Optional) Click *Custom Content Filter* to display the *Edit custom trace logging content filter* text box.

   The Custom Content Filter allows you to focus trace content on a specific section of the system where you suspect a problem exists. The filter is an XML document that specifies which trace logging content to send to the trace logger. You can limit the trace logging to one or more Java class files, or to one or more Java packages, or to one or more thread identifiers defined by Novell.

   4a Click *Default* to insert the default XML text.

   4b To validate this XML, the Java class or package must be completed.

   Knowledge of the Java class structure of the Access Manager product is required to create a Custom Content Filter. Therefore, it is recommended that this feature be used only with help from Novell Customer Support.

   For information about using the filter, see Appendix E, "Logging: Using the Custom Content Filter," on page 713.
5. To quickly trace content for specific parts of the system, select one of the following filters. The results are written to the file logger.

   Application: Logs system-wide trace content, except content that belongs to a specific protocol subsystem.

   Liberty: Logs trace content specific to the Liberty IDFF protocol and profiles.

   SAML 1: Logs trace content specific to the SAML 1.1 protocol and profiles.

   SAML 2: Logs trace content specific to the SAML 2 protocol and profiles.

   Web Service Provider: Logs trace content specific to fulfilling Web service requests from other Web service consumers.

   Web Service Consumer: Logs trace content specific to requesting Web services from a Web service provider.

Request/Response: Logs trace content specific to sending and receiving requests on all protocols, such as Liberty, SAML 1.1, and SAML 2.

User Stores: Logs trace content specific to accessing user stores. During a health check, the system includes all user stores in the configuration store.

Configuration: Logs trace content specific to configuring the system.

6. Click *OK*.
7. Update the Identity Server configuration (using Update Servers on the Servers page).
8. Restart the embedded service providers on the Access Gateways, in order to apply the changes.

   When you disable trace logging, you need to update the Identity Server configuration and restart the embedded service provides.

32.4 Configuring Access Gateway Logging

Logging HTTP transactions has associated costs. The Access Gateway is capable of handling thousands of transactions per second. If transaction volume is high and each log entry consumes a few hundred bytes, the Access Gateway can fill up the available disk space in a matter of minutes. HTTP logging also increases system overhead, which causes some degradation in performance. By default, the logging of HTTP transactions is turned off. Before enabling logging, you need to determine what needs to be logged and then plan a logging strategy.

- Section 32.4.1, "Determining Logging Requirements," on page 523
- Section 32.4.2, "Calculating Rollover Requirements," on page 524
- Section 32.4.3, "Enabling Logging," on page 526
- Section 32.4.4, "Configuring Common Log Options," on page 527
- Section 32.4.5, "Configuring Extended Log Options," on page 529
- Section 32.4.6, "Configuring Log Pushing," on page 532
- Section 32.4.7, "Configuring the Size of the Log Partition," on page 534

32.4.1 Determining Logging Requirements

Because logging requirements and transaction volume vary widely, Novell cannot make recommendations regarding a specific logging strategy. The following tasks guide you through the process of creating a strategy that fits your business needs.

1. Identify the reasons for tracking transactions such as customer billing, statistical analysis, or growth planning.
2. Determine which resources need logging.

   You enable logging at the proxy service level. If you have a proxy service protecting resources whose transactions do not need to be logged, reconfigure your proxy services so that the proxy service you configure for logging contains only the resources for which you want to log transactions.
3. Determine what information you need in each log entry.

   The common configuration for a log entry contains minimal information: the date, time, and client IP address for each entry. If you need more information, you can to select the extended log configuration. Do not select all available fields, but carefully select what you really need.

For example, you can include cookie information, but cookie information can consume a large amount of space and might not include any critical information you need.

You should log only the essential data because a few bytes can add up quickly when the Access Gateway is tracking thousands of hits every second. For information about what is available in an extended log profile, see Section 32.4.5, "Configuring Extended Log Options," on page 529.

4 Design a rollover strategy.

A log must be closed before it can be downloaded to another server for analysis or deleted. You specify either by time or size when the Access Gateway closes a log file and creates a new one. For each proxy service that you enable for logging, you need to reserve enough space for at least two files: one for logging and one for roll over. To calculate the best procedure, see Section 32.4.2, "Calculating Rollover Requirements," on page 524.

5 Design a log deletion strategy

The Access Gateway has a limited amount of disk space allocated for logging, and you need to decide how you are going to manage this space. You can limit the number of rollover files by number or age. You can also select to copy the files to another server and then delete them. To calculate the best procedure, see Section 32.4.2, "Calculating Rollover Requirements," on page 524.

32.4.2 Calculating Rollover Requirements

You can have the Access Gateway roll over log files based on time or on size, but not both. If you already know which option you want to use, scan this section and then complete only the calculations pertinent to your choice. If you don't know which option best matches your situation, completing the calculations in this section should help you decide.

The following variables are used in the formulas:

- **logpartition_size:** The total disk capacity reserved for log files on the Access Gateway.

  The Access Gateway reserves 4 GB to share between logging and system files. The system files do not grow significantly, so you can assume that you have about 2 GB for logging. To increase this size, see Section 32.4.7, "Configuring the Size of the Log Partition," on page 534.
- **logentry_size:** The average log entry size.

  You can determine this by configuring a proxy service to track the required information, generating traffic to the proxy service, downloading the log files, determining how large each entry is, and calculating the average.
- **request_rate:** The peak rate of requests per second.

  You can estimate this rate or place your Access Gateway in service and get more accurate data by accessing generated statistics. See Section 33.2, "Monitoring Access Gateway Statistics," on page 536.
- **num_services:** The number of proxy services for which you plan to enable logging.
- **logs_per_service:** The number of log files, both active and closed, that you want the Access Gateway to generate for each proxy service before the disk fills.

  You must plan to have at least two logs per proxy service, but you can have three or more.

The following formulas can help you estimate when the system would run out of resources:

- "Calculating diskfull_time" on page 525

- "Calculating max_roll_time" on page 525
- "Calculating max_log_roll_size" on page 526

Calculating diskfull_time

Using the following formula, you can calculate how long it will take the Access Gateway to fill your logging disk space:

```
diskfull_time in seconds = logpartition_size / (request_rate *
   logentry_size * num_services)
```

For example, assume the following:

logpartition_size = 1 GB (1,073,741,824 bytes)
request_rate = 1000 requests per second
logentry_size = 1 KB (1,024 bytes)
num_services = 1

```
diskfull_time = (1 GB) / (1000 * 1 KB * 1) = 1048 seconds (17.47
   minutes)
```

The logging disk space will fill up every 17.47 minutes.

To calculate the diskfull_time for your Access Gateway:

1. Determine the values of the four variables listed above.
2. Using the diskfull_time formula, calculate how often you can expect your logging disk to fill; then use the result in Calculating max_roll_time.

If your diskfull_time interval is too short to be practical for your rollover schedule, the easiest option is to reduce the log entry size by configuring the proxy services to log less information per transaction.

Calculating max_roll_time

Using the following formula, you can calculate the maximum rollover time value you should specify in the *Roll over every* field

```
max_roll_time = diskfull_time / logs_per_service
```

For example, assume the following:

diskfull_time = 12 hours
logs_per_service = 2

```
max_roll_time = 12 / 2 = 6 hours
```

If you roll your logs over by time intervals, the maximum time should be less than six hours. Otherwise, scheduling the download and deletion of log files is much more complicated and the window in which this can be done is narrower.

To calculate the max_roll_time for your Access Gateway:

1. Determine how many log files you want the Access Gateway to generate per service before log space fills.

   The minimum number is two.

2 Using the max_roll_time formula and the diskfull_time value obtained in "Calculating diskfull_time" on page 525, calculate how often you should have the cache device roll over the log files.

3 Record the max_roll_time result on your planning sheet.

Calculating max_log_roll_size

Using the following formula, you can calculate the maximum log file size you should specify in the *Maximum File Size* field:

```
max_log_roll_size = logpartition_size / (num_services *
   logs_per_service)
```

For example, assume the following:

logpartition_size = 600 MB
num_services = 2
logs_per_service = 3

```
max_log_roll_size = 600 MB / (2 * 3) = 100 MB
```

If you roll your logs over when they reach a specific size, the file size must be no more than 100 MB. Otherwise, the system runs out of disk space before you have three complete log files and scheduling the download and deletion of log files is much more complex.

To calculate the max_log_roll_size for your Access Gateway:

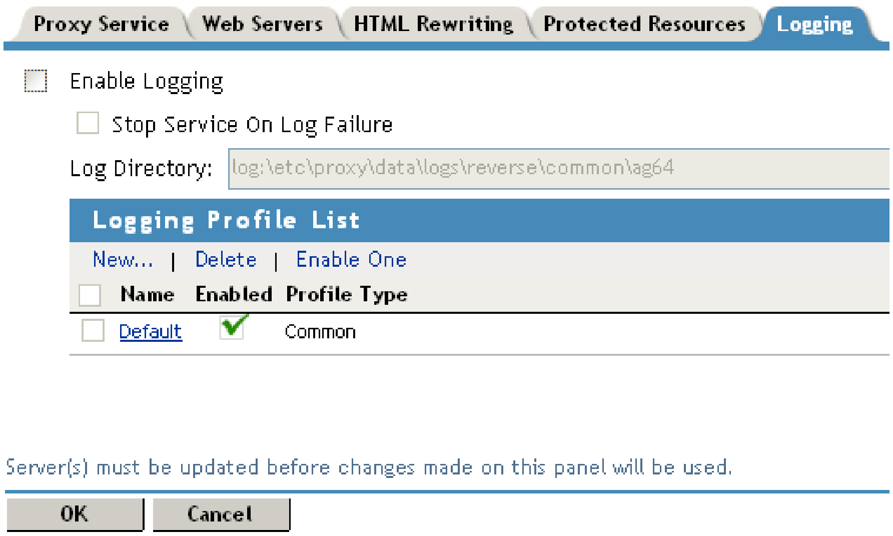
1 Determine the values of the three variables listed above.

2 Using the max_log_roll_size formula, calculate the maximum size a log file should reach before the cache device rolls it over.

32.4.3 Enabling Logging

Do not enable logging until you have designed a logging strategy. See Section 32.4.1, "Determining Logging Requirements," on page 523.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Logging*.



☐ Enable Logging

☐ Stop Service On Log Failure

Log Directory: log:\etc\proxy\data\logs\reverse\common\ag64

Logging Profile List

New... | Delete | Enable One

☐	Name	Enabled	Profile Type
☐	Default	✓	Common

Server(s) must be updated before changes made on this panel will be used.

2 Fill in the following fields:

Enable Logging: Select this field to enable logging.

Stop Service On Log Failure: Select this field if you want the Access Gateway to deny requests to this proxy service because the Access Gateway cannot log entries for it.

Log Directory: Displays the default location for the log files for this proxy service.

3 In the *Logging Profile List*, click one of the following options:

- **New:** Click this option to create a new logging profile. Then specify a name and select either *Common* or *Extended*.
- **Default:** Click *Default* to modify or view the settings for the *Default* profile. The *Default* profile uses the common log options.

A logging profile determines the type of information that is written to the log file; it also manages rollover and old file options.

4 Continue with one of the following:

- Section 32.4.4, “Configuring Common Log Options,” on page 527
- Section 32.4.5, “Configuring Extended Log Options,” on page 529

32.4.4 Configuring Common Log Options

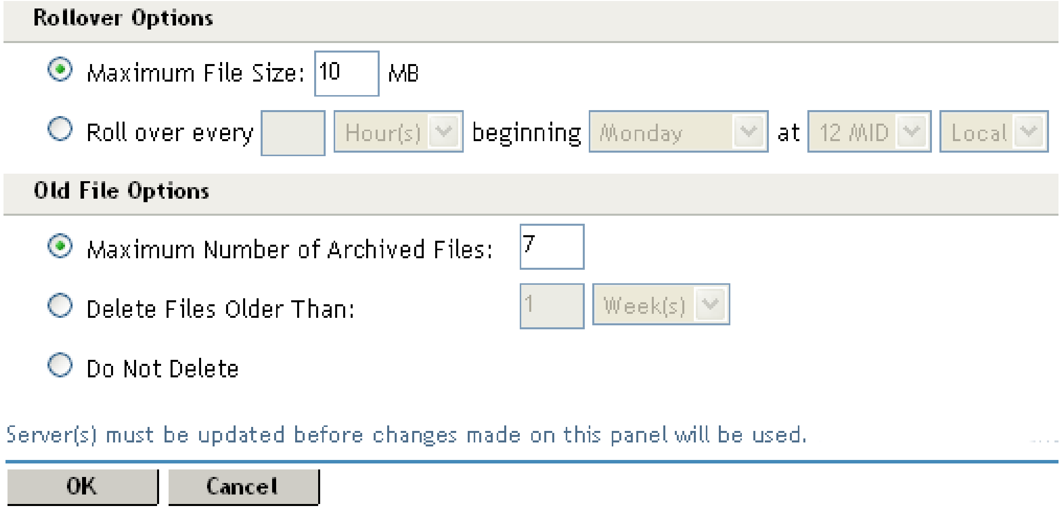
Use the common log options page to control log rollover and old file options. The data included in a log entry is controlled by a default configuration that includes the following:

- Date and time of the request
- Username of the client
- Remote host name
- The request line as it came from the client
- The HTTP status code returned to the client
- The number of bytes in the document transferred to the client

The Access Gateway does not allow active log files to be deleted. Only log files that have been closed can be deleted. The rollover options allow you to control when a file is rolled over and closed, and a new file is created. The old file options allow you to control when the rolled-over log files are deleted.

To configure a default log file for a selected proxy service:

1 Click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Logging* > *[Name of Common Log Profile]*.



Rollover Options

● Maximum File Size: 10 MB

○ Roll over every Hour(s) beginning Monday at 12 MID Local

Old File Options

● Maximum Number of Archived Files: 7

○ Delete Files Older Than: 1 Week(s)

○ Do Not Delete

Server(s) must be updated before changes made on this panel will be used.

2 Select one of the following roll over options:

Maximum File Size: Rolls the file when it reaches the specified number of megabytes.

Roll over every: Rolls the file at the specified interval. You can specify the interval in hours or days.

- **beginning:** Specifies the day that the interval should begin. You can select a day of the week or the first of the month.
- **at:** Select the hour of the day that the interval should begin and the time zone (either the local time zone or GMT).

3 Select one of the following old file options:

Maximum Number of Archived Files: Allows you to limit the number of old log files on the system to the number specified in this option. The oldest file is automatically deleted when this number is reached. All logging data in deleted files is lost. If you configure the *Log Push* option, you can set the system up so that the files are copied to another server before they are deleted from the server.

Delete Files Older Than: Allows you to configure the Access Gateway to delete files when they are older than the time you specify. All logging data in deleted files is lost. If you configure the *Log Push* option, you can set the system up so that the files are copied to another server before they are deleted from the server.

Do Not Delete: Prevents the system from automatically deleting the log files. You can use the *Log Push* option to copy the files to another server and then either manually delete them or have the *Log Push* option delete them from the server after they are copied to another server.

For information about the *Log Push* option, see Section 32.4.6, “Configuring Log Pushing,” on page 532.

4. Click *OK*.
5. Click the *Access Gateways* link, then click *Update* > *OK*.

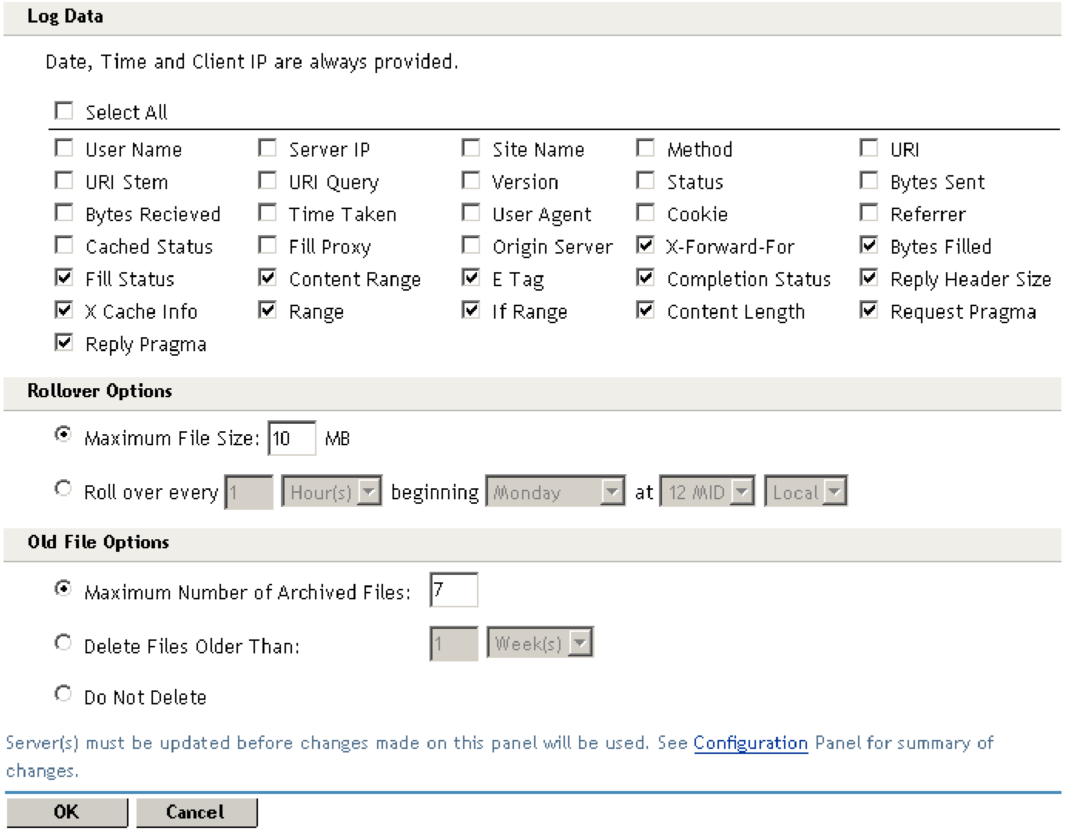
32.4.5 Configuring Extended Log Options

Use the extended log options page to control log entry content, log rollover, and old file options. A log entry always includes the date, time, and client IP address for each entry, but with the log data options, you can add other fields such as the IP address of the server and the username of the client.

The Access Gateway does not allow active log files to be deleted. Only log files that have been closed can be deleted. The rollover options allow you to control when a file is rolled over and closed, and a new file is created. The old file options allow you to control when the rolled-over log files are deleted.

To configure an extended log file for a selected proxy service:

1. Click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Logging* > *[Name of Extended Log Profile]*.



Log Data

Date, Time and Client IP are always provided.

☐ Select All

☐ User Name
☐ URI Stem
☐ Bytes Recieved
☐ Cached Status
☑ Fill Status
☑ X Cache Info
☑ Reply Pragma
☐ Server IP
☐ URI Query
☐ Time Taken
☐ Fill Proxy
☑ Content Range
☑ Range
☐ Site Name
☐ Version
☐ User Agent
☐ Origin Server
☑ E Tag
☑ If Range
☐ Method
☐ Status
☐ Cookie
☑ X-Forward-For
☑ Completion Status
☑ Content Length
☐ URI
☐ Bytes Sent
☐ Referrer
☑ Bytes Filled
☑ Reply Header Size
☑ Request Pragma

Rollover Options

● Maximum File Size: 10 MB

○ Roll over every 1 Hour(s) beginning Monday at 12 MID Local

Old File Options

● Maximum Number of Archived Files: 7

○ Delete Files Older Than: 1 Week(s)

○ Do Not Delete

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2. Select one or more of the log data options:

Name	Description
User Name	The name of the user sending the request.
Server IP	The IP address of the Access Gateway.
Site Name	The name of the reverse proxy.
Method	The HTTP method the browser sent to the Access Gateway.
URI	The HTTP URL the browser sent to the Access Gateway.
URI Stem	The stem portion of the HTTP URL the browser sent to the Access Gateway. The stem is everything in the URL up to the first question mark. If the URL has no question mark, the *URI Stem* field is the same as the *URI* field. It is redundant if *URI* is selected.
URI Query	The query portion of the HTTP URL the browser sent to the Access Gateway. The query is everything from the first question mark through the end of the URL. If the URL has no question mark, this field has no value. It is redundant if URI is selected.
Version	The HTTP version specified in the URL the browser sent to the Access Gateway.
Status	The HTTP status code the Access Gateway sent to the browser.
Bytes Sent	The number of bytes of HTTP response data the Access Gateway sent to the browser.
Bytes Received	The number of bytes of HTTP request data the proxy service received from the browser.
Time Taken	The time in seconds it took the Access Gateway resources to deal with the request.
U*ser Agent*	The User-Agent HTTP request header value the browser sent to the Access Gateway.
Cookie	The Cookie HTTP request header value the browser sent to the Access Gateway. The Access Gateway doesn't cache cookie information. Cookies can consume a lot of space. If you select this option, make sure it contains the critical information that you need.
Referer	The Referer HTTP request header value the browser sent to the Access Gateway.
Cached Status	The value indicates whether the request was filled from cache. 1 = filled from cache 0 = not filled from cache
Fill Proxy	The IP address of the upstream proxy.
Origin Server	The IP address of the Web server. This assumes the Access Gateway retrieved the requested information directly from the Web server.
X-Forward-For	The X-Forwarded-For HTTP request header value the browser sent to the Access Gateway. Do not confuse this with the X-Forwarded-For option that causes the Access Gateway to generate or forward headers to upstream proxies or Web servers.

Name	Description
Bytes Filled	(Linux only) The total bytes filled in response to the request.
Fill Status	(Linux only)
Content Range	(Linux only) The byte ranges sent from the Access Gateway to a requesting browser.
E Tag	(Linux only) The tag sent from the Access Gateway to a requesting browser.
Completion Status	(Linux only) The completion status for the transaction indicating that it completed successfully or that it failed. Possible values: success, timeout, reset (the client terminated the connection), administrative (the Access Gateway terminated the connection).
Reply Header Size	(Linux only) The size in bytes of the HTTP header associated with a response to a client.
X Cache Info	(Linux only) Brief status statement for cached objects; brief reasons why an object was not cached.
Range	(Linux only) The Range header value.
If Range	(Linux only) The If Range header value, which indicates whether the browser request was a conditional range request.
Content Length	(Linux only) The size in bytes of the entire object delivered to a requesting browser.
Request Pragma	(Linux only) The pragma value associated with a browser request.
Reply Pragma	(Linux only) The pragma value associated with a server response to a requesting browser.

3. Select one of the following rollover options:

   Maximum File Size: Rolls the file when it reaches the specified number of megabytes.

   Roll over every: Rolls the file at the specified interval. You can specify the interval in hours or days.

   - **beginning:** Specifies the day that the interval should be begin. You can select a day of the week or the first of the month.
   - **at:** Select the hour of the day that the interval should begin and the time zone (either the local time zone or GMT).

4. Select one of the following old file options.

   Maximum Number of Archived Files: Allows you to limit the number of old log files on the system to the number specified in this option. The oldest file is automatically deleted when this number is reached. All logging data in deleted files is lost. If you configure the Log Push option, you can set the system up so that the files are copied to another server before they are deleted from the server.

   Delete Files Older Than: Allows you to configure the Access Gateway to delete files when they are older than the time you specify. All logging data in deleted files is lost. If you configure the *Log Push* option, you can set the system up so that the files are copied to another server before they are deleted from the server.

Do Not Delete: Prevents the system from automatically deleting the log files. You can use the *Log Push* option to copy the files to another server and then either delete them manually or have the *Log Push* option delete them from the server when they have been copied to another server.

For information about the *Log Push* option, see Section 32.4.6, "Configuring Log Pushing," on page 532.

5 Click *OK*.

6 Click the *Access Gateways* link, then click *Update* > *OK*.

32.4.6 Configuring Log Pushing

(NetWare only) The *Log Push* option allows you to configure the NetWare Access Gateway to copy log files to an FTP server at specified intervals. The *Log Push* option is configured for all log files on the Access Gateway. If you have enabled logging on multiple proxy services, the Access Gateway uses the same configuration to push the log files of each proxy service.

This feature works within the following parameters:

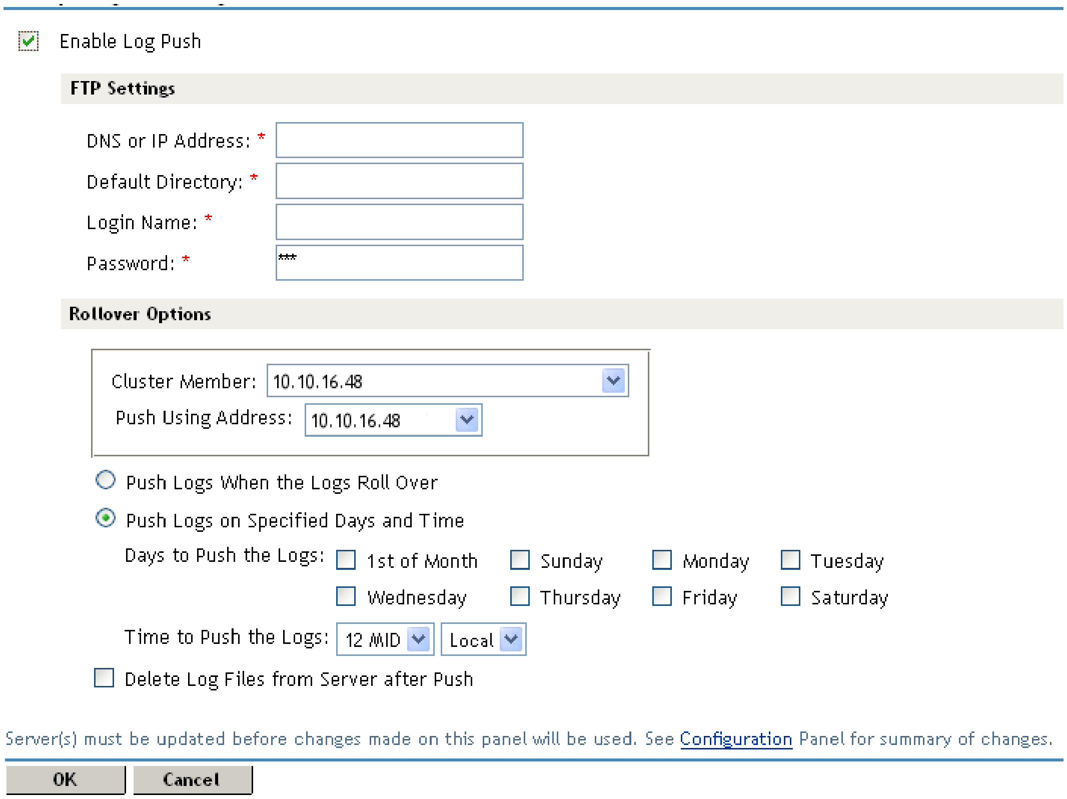
- The Access Gateway tries as many times as necessary to establish a connection with the FTP server during the hour of the scheduled push. When the hour changes, the Access Gateway stops trying until the next interval you have specified.
- When the connection is established, the Access Gateway assumes that pushing the log files was successful. The Access Gateway does not detect any errors that prevent the successful pushing of the files.

For example, you specify that log files are to be pushed on every day of the week at 12 midnight. When the system clock reaches the target hour, the Access Gateway begins trying to establish a connection with the FTP server.

- If a connection cannot be established before the hour changes to 1 a.m., the Access Gateway stops trying to connect and doesn't try again until 12 midnight the next day.
- If a connection is established but an error occurs that prevents a successful push, the error is not detected, and the Access Gateway doesn't try to connect again until 12 midnight the next day.

To configure log pushing:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Log Push*.



☑ Enable Log Push

FTP Settings

DNS or IP Address: *

Default Directory: *

Login Name: *

Password: * ***

Rollover Options

Cluster Member: 10.10.16.48

Push Using Address: 10.10.16.48

○ Push Logs When the Logs Roll Over

● Push Logs on Specified Days and Time

Days to Push the Logs: ☐ 1st of Month ☐ Sunday ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday ☐ Saturday

Time to Push the Logs: 12 MID Local

☐ Delete Log Files from Server after Push

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2 To enable log pushing, select *Enable Log Push*.

3 Configure the following FTP settings. All of them are required settings.

DNS or IP Address: Specify the DNS name or the IP address of your FTP server.

Default Directory: Specify the directory on the FTP server to which the Access Gateway should copy the log files.

Login Name: Specify the name that the Access Gateway should use to log in to the FTP server.

Password: Specify the password that the Access Gateway should use for logging in.

4 To schedule when the log files are copied to the FTP server, fill in the following fields:

Cluster Member: (Available only if the Access Gateway is a member of a cluster.) Select the server you want to configure from the list of servers. The modifications made to the *Push Using Address* option apply only to the selected cluster member. Modifications made to any other options on the page apply to all members of the cluster.

Push Using Address: Select the IP address you want to use for sending the log files to the FTP server.

5 Select when you want the logs to be pushed. Select one of the following:

- **Push Logs when the Logs Roll Over:** To push the logs as soon as a log file rolls over, select *Push Logs when the Logs Roll Over*.This method ensures that log files are copied as soon as possible.
- **Push Logs on Specified Days and Time:** To push the logs on selected days at a specific time, select *Push Logs on Specified Days and Time*, then configure the following fields:

  Days to Push the Logs: Allows you to select the days when the log push should occur. You can select multiple days for pushing.

Time to Push the Logs: Specifies the time of day when the log files are pushed.

6 Specify what you want done with the log files after they have been copied to the FTP server.

Select the *Delete Log Files from Server after Push* option to have the Access Gateway delete the log files after they have been copied to the FTP server. This is the recommended method. If you do not select this option, you must manually delete them or use the old file options on the Logging page (see Section 32.4.4, “Configuring Common Log Options,” on page 527).

7 Click *OK*.

8 Click the *Access Gateways* link, then click *Update* > *OK*.

32.4.7 Configuring the Size of the Log Partition

The size of the log partition should be configured as part of the installation process. See one of the following in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*:

- The NetWare Access Gateway creates a 2 GB log: volume. To increase its size, see “Configuring the Log Partition on the NetWare Access Gateway ”
- Linux Access Gateway logs are stored in `/root` partition by default. You can create a `/var` partition to store the logs. The size of this partition depends on your requirements. For more information on creating the `/var` partition, see “Customizing the Partitions” in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

Viewing Statistics

33

Statistics can indicate that the system is functioning optimally or that it has some bottlenecks.

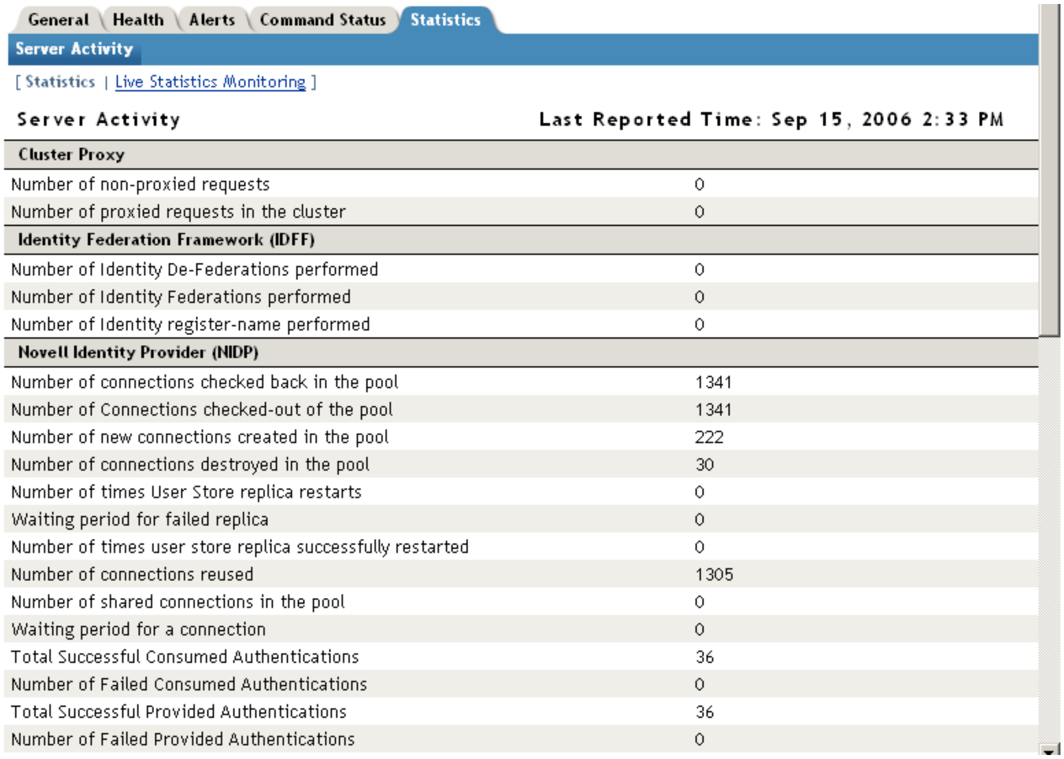
- Section 33.1, “Monitoring Identity Server Statistics,” on page 535
- Section 33.2, “Monitoring Access Gateway Statistics,” on page 536
- Section 33.3, “Viewing SSL VPN Statistics,” on page 547

33.1 Monitoring Identity Server Statistics

The Statistics page allows you to monitor the amount of data and the type of data the Identity Server is processing. You can specify the intervals for the refresh rate and, where allowed, view graphic representations of the activity.

1 In the Administration Console, choose *Access Manager > Identity Servers*.

2 In the Statistics column, click *View*.



Server Activity — Last Reported Time: Sep 15, 2006 2:33 PM

Cluster Proxy	
Number of non-proxied requests	0
Number of proxied requests in the cluster	0
Identity Federation Framework (IDFF)	
Number of Identity De-Federations performed	0
Number of Identity Federations performed	0
Number of Identity register-name performed	0
Novell Identity Provider (NIDP)	
Number of connections checked back in the pool	1341
Number of Connections checked-out of the pool	1341
Number of new connections created in the pool	222
Number of connections destroyed in the pool	30
Number of times User Store replica restarts	0
Waiting period for failed replica	0
Number of times user store replica successfully restarted	0
Number of connections reused	1305
Number of shared connections in the pool	0
Waiting period for a connection	0
Total Successful Consumed Authentications	36
Number of Failed Consumed Authentications	0
Total Successful Provided Authentications	36
Number of Failed Provided Authentications	0

3 Click either of the following options:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

4 Review the following statistics:

- Cluster Proxy
- Identity Federation Framework
- Novell Identity Provider
- SAML
- SAML 2
- Web Services Framework

5 Click *Close* to return to the Servers page.

33.2 Monitoring Access Gateway Statistics

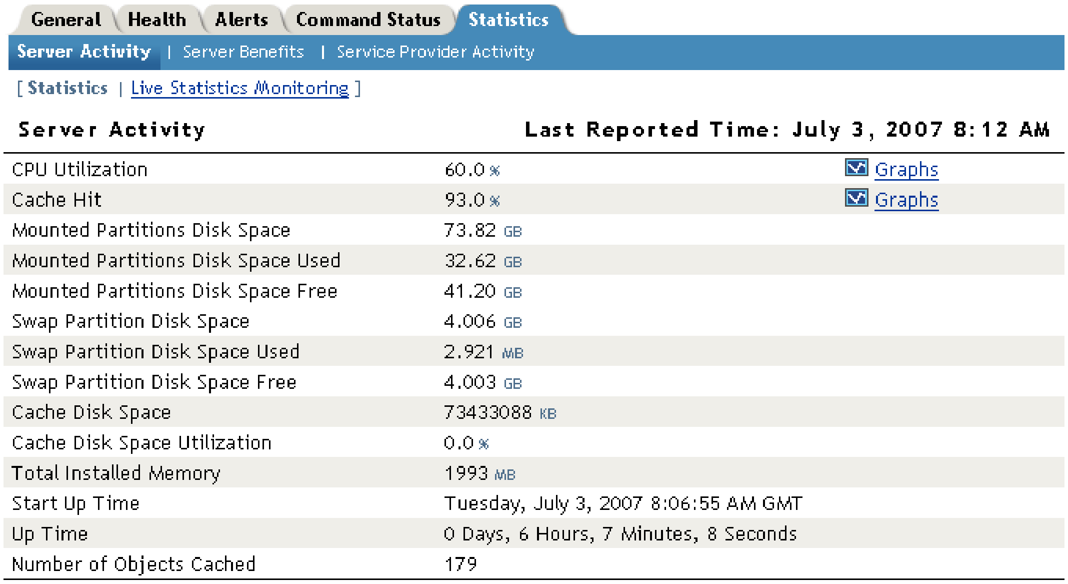
Access Gateway statistics are available for each Access Gateway and for clusters:

- Section 33.2.1, "Viewing Access Gateway Statistics," on page 536
- Section 33.2.2, "Viewing Cluster Statistics," on page 546

33.2.1 Viewing Access Gateway Statistics

The Statistics page allows you to monitor the amount of data and the type of data the Access Gateway is processing.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Server]* > *Statistics*.



Server Activity	**Last Reported Time: July 3, 2007 8:12 AM**	
CPU Utilization	60.0 %	Graphs
Cache Hit	93.0 %	Graphs
Mounted Partitions Disk Space	73.82 GB	
Mounted Partitions Disk Space Used	32.62 GB	
Mounted Partitions Disk Space Free	41.20 GB	
Swap Partition Disk Space	4.006 GB	
Swap Partition Disk Space Used	2.921 MB	
Swap Partition Disk Space Free	4.003 GB	
Cache Disk Space	73433088 KB	
Cache Disk Space Utilization	0.0 %	
Total Installed Memory	1993 MB	
Start Up Time	Tuesday, July 3, 2007 8:06:55 AM GMT	
Up Time	0 Days, 6 Hours, 7 Minutes, 8 Seconds	
Number of Objects Cached	179	

2 Select from the following types:

- "Server Activity" on page 537
- "Server Benefits" on page 540
- "Server SSL Activity" on page 541 (NetWare only)
- "Top 20 Sites" on page 542 (NetWare only)

  - "Service Provider Activity" on page 542
  - "Configured Services" on page 546 (NetWare only)

3 Click *Close*.

Server Activity

Access Gateways > [Name of Server] > Statistics

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

These general statistics are grouped into the following categories:

- Server Activity
- Connections
- Bytes
- Requests
- Cache Freshness

Server Activity

The Server Activity section displays general server utilization statistics.

Table 33-1 *Server Activity*

Statistic	Description
CPU Utilization	Displays the current CPU utilization rate. Use the available graph for capacity planning.
Cache Hit	Displays the current cache hit rate. A high cache hit rate indicates that the caching system is off-loading significant request processing from the Web servers whose objects have been cached. Use the available graph for capacity planning.
Mounted Partitions Disk Space	(Linux only) Displays the total disk space configured on mounted partitions.
Mounted Partitions Disk Space Used	(Linux only) Displays the disk space in use on mounted partitions.
Mounted Partitions Disk Space Free	(Linux only) Displays the disk space available on mounted partitions.
Swap Partition Disk Space	(Linux only) Displays the total disk space configured for the swap partition.
Swap Partition Disk Space Used	(Linux only) Displays the disk space in use on the swap partition.

Statistic	Description
Swap Partition Disk Space Free	(Linux only) Displays the disk space available on the swap partition.
Cache Disk Space	Displays the total disk space available for caching. The amount shown is smaller than the total disk space available on the Access Gateway because it doesn’t include the disk space reserved for the operating system and for log files.
Cache Disk Space Utilization	Displays the percentage of caching disk space currently in use.
Total Installed Memory	Displays the amount of memory that is installed on the Access Gateway.
Start Up Time	Displays the last time the Access Gateway was started.
Up Time	Displays the total time the Access Gateway has been running since it was last started.
Number of Objects Cached	Displays the total number of Web objects that have been cached.

Connections

The connection statistics show the current and peak levels of usage in terms of TCP connections.

Table 33-2 *Connections*

Statistic	Description
Current Connections to Origin Server	Displays the current number of connections that the Access Gateway has established with Web servers.
Current Connections to Browsers	Displays the current number of connections that the Access Gateway has established with browsers.
Current Total Connections	Displays the current total of all connections that the Access Gateway has established.
Connections to Origin Server	Displays the total number of connections that the Access Gateway has established with Web servers since it was last started.
Peak Connections from Origin Server	Displays the peak number of connections that the Access Gateway has established with Web servers.
Connections to Browsers	Displays the total number of connections that the Access Gateway has established with browsers since it was last started.
Peak Connections to Browsers	Displays the peak number of connections that the Access Gateway has established with browsers.
Total Connections through SOCKS	Displays the total number of connections the Access Gateway has established through a firewall.
Failed Connection Attempts	Displays the total number of failed connection attempts the Access Gateway has made while attempting to fill its Web object cache.

Bytes

The bytes statistics show how fast information is being sent in response to the following types of requests:

- Browser requests to the Access Gateway
- Access Gateway requests to the Web servers

Table 33-3 *Bytes*

Statistic	Description
Bytes per Second from Origin Server	Displays the number of bytes of data being sent each second from the Web servers to the Access Gateway.
Bytes per Second to Browsers	Displays the number of bytes of data being sent each second from the Access Gateway to the browsers.
Total Bytes per Second	Displays the total number of bytes of data being sent each second from the Access Gateway and from the Web servers.
Bytes Received from Origin Server	Displays the total number of bytes of data sent to the Access Gateway from the Web servers since the Access Gateway last started.
Bytes Sent to Browser	Displays the total number of bytes of data sent to the browsers from the Access Gateway since the Access Gateway last started.
Total Bytes	Displays the total number of bytes of data sent from the Access Gateway and from the Web servers since the Access Gateway was last started.

Requests

The request statistics show the number of requests that are being sent from the browsers to the Access Gateway and from the Access Gateway to the Web servers.

Table 33-4 *Requests*

Statistic	Description
Current Requests to Origin Server	Displays the current number of requests that the Access Gateway has made to the Web servers.
Current Requests from Browsers	Displays the current number of requests that the browsers have made to the Access Gateway.
Total Current Requests	Displays the total number of current requests that the Access Gateway has received from the browsers and that the Access Gateway has sent to the Web servers.
Successful Requests to Origin Server	Displays the total number of successful requests that the Access Gateway has sent to the Web servers since the Access Gateway last started.
Failed Requests to Origin Server	Displays the total number of failed requests that the Access Gateway has sent to the Web servers since the Access Gateway last started.
Cumulative Requests to Origin Server	Displays the total number of requests that the Access Gateway has sent to the Web servers since the Access Gateway last started.

Statistic	Description
Cumulative Requests to Browsers	Displays the total number of requests that the browsers have sent to the Access Gateway since the Access Gateway last started.
Total Cumulative Requests	Displays the total number of cumulative requests that the Access Gateway has processed since the Access Gateway last started.
Requests per Second to Origin Server	Displays the number of requests that are being sent each second from the Access Gateway to the Web servers.
Requests per Second from Browsers	Displays the number of requests that are being sent each second from the browsers to the Access Gateway.
Total Requests per Second	Displays the total number of requests that are being sent each second from the Access Gateway and from the browsers.
Peak Requests per Second to Origin Server	Displays the peak number of requests that have been sent in one second from the Access Gateway to the Web servers.
Peak Requests per Second from Browsers	Displays the peak number of requests that have been sent in one second from the browsers to the Access Gateway.

Cache Freshness

The cache freshness statistics display information about the cache refresh process.

Table 33-5 *Cache Freshness*

Statistic	Description
Total “Get If Modified Since” Request	Displays the total number of Get If Modified Since requests that the Access Gateway has received from browsers.
Total Not Modified Replies	Displays the total number of 304 Not Modified replies that the Access Gateway has received from the Web servers for updated content.
Cache Freshness	Displays the percentage of objects in cache that are considered fresh.
Oldest Object in Memory	Displays how long the oldest cache object has been cached.

Server Benefits

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

The Server Benefits page displays information about bandwidth and DNS caching:

Table 33-6 *Server Benefits*

Statistic	Description
Total Bandwidth Saved	(Linux only) Displays the amount of bandwidth saved by using data cached by the Access Gateway rather than requesting the data from the Web servers.
Bytes Saved per Second	Displays how many bytes of data the Access Gateway was able to send from cache rather than requesting it from the Web servers.
Bandwidth Saved	Displays the amount of bandwidth saved by using data cached by the Access Gateway rather than requesting the data from the Web servers.
Total DNS Lookups Saved	(Linux only) Displays the number of DNS requests that the Access Gateway could solve locally without performing a DNS lookup.
DNS “Modified Since” Queries Returning False	(Linux only) Displays the number of DNS Modified Since queries that the Access Gateway was able to service with a false value.
Total Number of Connections Saved	Displays the number of connections that the Access Gateway has with clients minus the number of connections that the Access Gateway has with Web servers. This statistic indicates the number of connections that the Access Gateway is off loading from the Web servers.

Server SSL Activity

(NetWare only) Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

The Server SSL Activity page displays information about the SSL communication process between the browsers and the Access Gateway and between the Access Gateway and the Web servers.

Table 33-7 *Server SSL Activity*

Statistic	Description
Current SSL Connection	Displays the current number of connections that are SSL connections.
Peak SSL Connections	Displays the peak number of SSL connections that the Access Gateway has established since it was last started.
Browser Full SSL Handshakes	Displays the number of browser connection requests that required a full handshake because the handshake information was not available from cache.
Browser Abbreviated Handshakes	Displays the number of browser connections requests that could perform an abbreviated handshake because the handshake information was still available from cache.
Browser SSL Alerts	Displays the number of SSL alerts that the browsers have sent to the Access Gateway.

Statistic	Description
Server SSL Full Handshakes	Displays the number of Web server connection requests that required a full handshake because the handshake information was not available from cache.
Server SSL Abbreviated Handshakes	Displays the number of Web server connection requests that could perform an abbreviated handshake because the handshake information was still available from cache.
Server SSL Alerts	Displays the number of SSL alerts that the Access Gateway sent to the Web servers.

Top 20 Sites

(NetWare only) Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

The Top 20 Sites page displays the DNS name or IP address of the sites that are receiving the most traffic. These sites are sorted by the number of hits they received and by the number of bytes of data they sent.

Service Provider Activity

Select whether to monitor live or static statistics:

Statistics: Select this option to view the statistics as currently gathered. The page is static and the statistics are not updated until you click *Live Statistics Monitoring*.

Live Statistics Monitoring: Select this option to view the statistics as currently gathered and to have them refreshed at the rate specified in the *Refresh Rate* field.

The ESP Activity page displays information about the communication process between the Access Gateway module (ESP) and the Identity Server. These statistics are grouped into the following categories:

- Cluster Proxy Statistics
- Identity Federation Framework Statistics
- Identity Provider Statistics
- SAML Statistics
- SAML 2 Statistics
- Web Service Framework Statistics

Cluster Proxy Statistics

Statistic	Description
Number of non-proxied requests	The total number of times the L4 switch sent the request to the server that established the session for the user making the request. When this happens, the server does not need to proxy the request to a peer cluster member.
Number of proxied request in the cluster	The total number of times a cluster member has determined that it did not establish the session for the user making the request and then proxied the request to the peer cluster member that did establish the session.

Identity Federation Framework Statistics

Statistic	Description
Number of Identity De-Federations performed	The number of requests to defederate user accounts that the Identity Server has processed.
Number of Identity Federations performed	The number of requests to federate user accounts that the Identity Server has processed.
Number of Identity register-name performed	The total number of register name requests that the Identity Server has processed.

Identity Provider Statistics

Statistic	Description
Number of connections checked back into the pool	The total number of times a user store connection has been checked into a connection pool after being checked out and used.
Number of connections checked out of the pool	The total number of times a user store connection has been checked out of a connection pool and used.
Number of new connections created in the pool	The total number of times the Identity Server has created a new connection to a user store.
Number of connections destroyed in the pool	The total number of times the Identity Server has destroyed a connection to a user store.
Number of times User Store replica restarts	When the Identity Server loses a connection to an LDAP user store, that user store is placed on a restart thread. After a period of time, the restart thread attempts to reconnect to the user store. This count is the total number of times that the user stores have been placed on the restart thread.
Waiting period for failed replica	The total number of times the restart thread has failed to regain a connection with a user store and has needed to wait the given time period before trying again.
Number of times user store replica successfully restarted	The total number of times the restart thread has successfully regained a connection with a user store.

Statistic	Description
Number of connections reused	The total number of times a user store connection has been reused. This means that the Identity Server was able to check out a connection from the pool and use an existing connection.
Number of shared connections in the pool	Each user store has two connection pools: a user pool and an admin pool. As connections are checked out of each of these pools, it might become apparent to the Identity Server that one pool is overworked and the other pool has unused connections. When this situation is detected, a connection is shared from one pool to the other. Thus, the admin pool might gain a connection and the user pool might lose one. This is the total number of times that connections have been shared (over all user stores).
Waiting period for a connection	The total number of times that all connections have been checked out, and the requesting thread has waited for a connection to become available.
Total successful consumed authentications	The number of successful logins that the Identity Server has processed.
Number of failed consumed authentications	The total number of failed logins that the Identity Server has processed (for any reason).
Total successful provided authentications	The number of successful authentications that the Identity Server has provided to other service providers, including embedded service providers.
Number of failed provided authentications	The number of failed authentications that the Identity Server has provided to other service providers, including embedded service providers.
Number of logouts	The total number of logout requests that the Identity Server has processed.
% of free memory	The current percentage of system memory that Java considers free.
Number of users currently logged in	The number of sessions that are currently active, which equates with the number of currently logged-in users.
Total requests	The total number of requests that have passed through the Identity Server.

SAML Statistics

Statistic	Description
Number of SAML requests	The total number of SAML1.1 query attribute requests that the Identity Server has processed.

SAML 2 Statistics

Statistic	Description
Number of SAML-2 Defederations	The total number of SAML-2 defederation requests that the Identity Server has processed.
Number of SAML-2 Federations	The total number of SAML-2 federation requests that the Identity Server has processed.

Statistic	Description
Number of SAML-2 requests	The total number of SAML-2 query attribute requests that the Identity Server has processed.
Number of SAML-2 register name	The total number of SAML-2 register name requests that the Identity Server has processed.

Web Service Framework Statistics

Statistic	Description
Number of credential-profile service ‘modify’	The total number of modify requests made to the Novell® Credential Profile Web Service.
Number of credential-profile service ‘query’	The total number of query requests made to the Novell Credential Profile Web Service.
Number of discovery service 'modify’	The total number of modify requests made to the Discovery Web Service.
Number of discovery service 'query'	The total number of query requests made to the Discovery Web Service.
Number of employee-profile service 'modify’	The total number of modify requests made to the Employee Profile Web Service.
Number of employee-profile service 'query'	The total number of query requests made to the Employee Profile Web Service.
Number of custom-profile service 'modify’	The total number of modify requests made to the Novell Custom Profile Web Service.
Number of custom-profile service 'query'	The total number of query requests made to the Novell Custom Profile Web Service.
Number of personal-profile service 'modify'	The total number of modify requests made to the Personal Profile Web Service.
Number of personal-profile service 'query'	The total number of query requests made to the Personal Profile Web Service.
Number of role-profile service 'modify ‘	The total number of modify requests made to the Novell Role Profile Web Service.
Number of role-profile service 'query'	The total number of query requests made to the Novell Role Profile Web Service.
Number of interaction service redirects by web services consumer (client)	The total number of times the Identity Server has been redirected to perform user interaction by using the User Interaction Redirection profile.
Number of interaction service redirects to server	The total number of times the Identity Server has handled a user interaction request that it received through the User Interaction Redirection profile.
Number of interaction service redirects initiated by web services consumer (client)	The total number of times the Identity Server has called a Trusted User Interaction Service by using the Trusted User Interaction Service profile.

Statistic	Description
Number of interaction service redirects handled by trusted server	The total number of times the Identity Server has handled a user interaction request that it received through the Trusted User Interaction Service profile.

Configured Services

(NetWare only) The Configured Services page displays information about all the services that have been configured on the selected Access Gateway. It includes information about the proxy services that you have configured as well as the system proxy services (the nesp and soapbc path-based proxy services).

33.2.2 Viewing Cluster Statistics

To view general performance statistics of the servers assigned to the selected cluster:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Cluster]* > *Statistics*.

2 To determine performance, analyze the following statistics:

Column	Description
Server Name	Lists the name of the Access Gateways that belong to the group. To view additional statistical information about a specific Access Gateway, click the name of an Access Gateway.
CPU %	Displays the current CPU utilization rate. Use this statistic for capacity planning.
Cache Hit Rate %	Displays the current cache hit rate. A high cache hit rate indicates that the caching system is off-loading significant request processing from the Web server whose objects have been cached. If the percentage is low, you might want to configure a pin list. For this and other caching options, see Chapter 16, “Configuring the Cache Settings,” on page 279.
Bytes per second to/from Server	Displays the rate at which the Access Gateway is requesting Web objects from the Web servers it is protecting.
Bytes per second to/from Browser	Displays the rate at which browser clients are requesting Web objects.
Current Connections	Displays the total number of TCP™ connections that are active, idle, or closing.
Statistics	Allows you to view all the statistics for a selected server. Click *View* to see these additional statistics. For more information, see Section 33.2, “Monitoring Access Gateway Statistics,” on page 536.

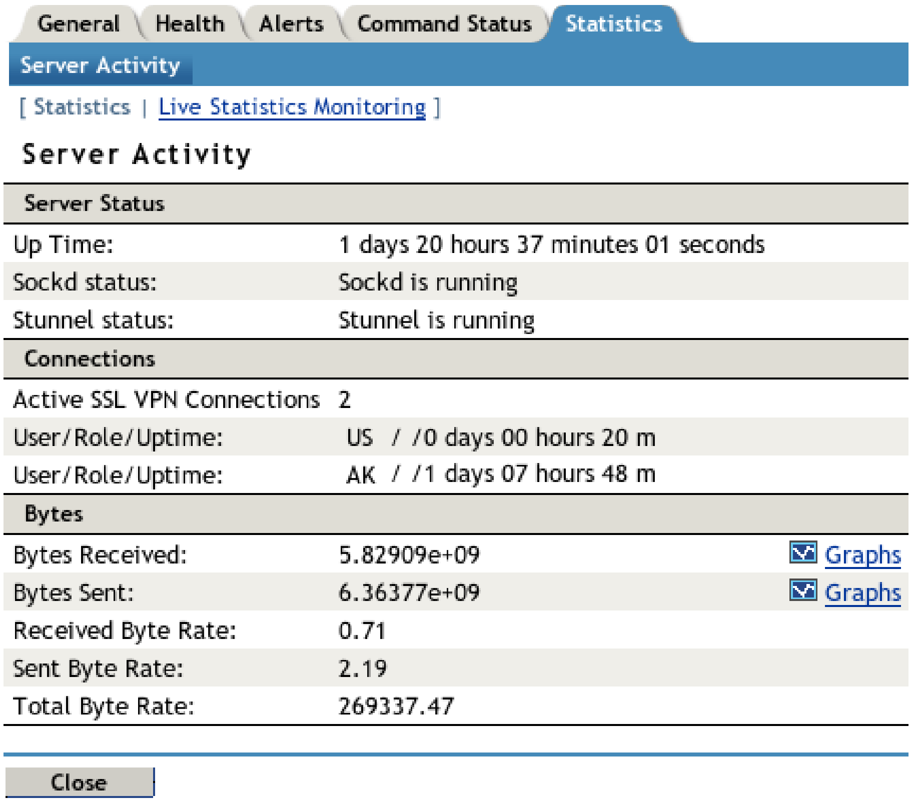
3 Click *Close*.

33.3 Viewing SSL VPN Statistics

The Statistics page allows you to view such information as the number of active client connections and the time when the SSL VPN server was started.

1 In Administration Console, click *Access Manager > SSL VPNs > [Server Name] > Statistics*.

The Server Statistics page is displayed.



Server Activity

Server Status		
Up Time:	1 days 20 hours 37 minutes 01 seconds	
Sockd status:	Sockd is running	
Stunnel status:	Stunnel is running	
Connections		
Active SSL VPN Connections	2	
User/Role/Uptime:	US / /0 days 00 hours 20 m	
User/Role/Uptime:	AK / /1 days 07 hours 48 m	
Bytes		
Bytes Received:	5.82909e+09	Graphs
Bytes Sent:	6.36377e+09	Graphs
Received Byte Rate:	0.71	
Sent Byte Rate:	2.19	
Total Byte Rate:	269337.47	

Server Status information is gathered in the following sections:

Column	Description
Up Time	Specifies the duration for which the server has been up and running.
Sockd Status	Specifies if the sockd is running or not.
Stunnel Status	Specifies if the Stunnel is running or not.

Connection information is gathered in the following sections:

Column	Description
Active SSL VPN Connections	Specifies the number of active SSL VPN connections. Username, role of the user, and uptime of each user are specified for each active connection.

Bytes information is gathered in the following sections:

Column	Description
Bytes Received	Specifies the number of bytes received. You can also view a graph, which lists the number of bytes sent for fixed intervals. For more information, see Viewing the Bytes Graphs.
Bytes Sent	Specifies the number of bytes sent. You can also view a graph, which lists the number of bytes sent for fixed intervals. For more information, see Viewing the Bytes Graphs.
Received Byte Rate	Specifies the percentage of bytes received.
Sent Byte Rate	Specifies the percentage of bytes sent.
Total Byte Rate	Specifies the total percentage of bytes transferred.

2 Select one of the following options:

- **Statistics:** To display the number of active client connections and the time when the server was started, click *Statistics*.
- **Live Statistics Monitoring:** To refresh the above information for a specified interval, click *Live Statistics Monitoring*. You can select the refresh interval from the *Refresh Rate* drop-down list.

3 Click *Close* to close the *Statistics* tab.

33.3.1 Viewing the Bytes Graphs

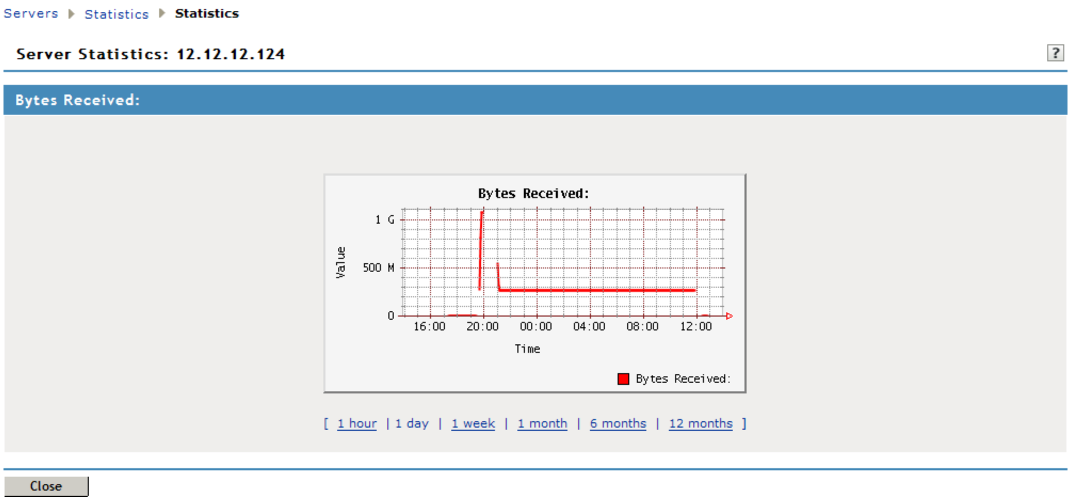
The number of bytes sent and bytes received can be viewed in the form of graphs. You can view graphs for the following time frames:

- **1 Hour:** The number of bytes sent or received every ten minutes.
- **1 Day:** The number of bytes sent or received every four hours.
- **1 Week:** The number of bytes sent or received every day.
- **1 Month:** The number of bytes sent or received every week.
- **6 Months:** The number of bytes sent or received every month for six months.
- **12 Months:** The number of bytes sent or received every month for one year.

To view graphs:

1 In Access Manager select, click *Access Manager > SSL VPNs > [Server Name] > Statistics.*

2 Select *Graphs* from either the *Bytes Received* or *Bytes Sent* section, depending on your needs.


Servers ▶ Statistics ▶ Statistics
Server Statistics: 12.12.12.124
?
Bytes Received:
Bytes Received:
Value
1 G
500 M
0
16:00
20:00
00:00
04:00
08:00
12:00
Time
🟥 Bytes Received:
[1 hour | 1 day | 1 week | 1 month | 6 months | 12 months]
Close

3 Click *Close* to close the Graphs page.

Managing Server Health

34

You can monitor all of the components hosted by a server and quickly isolate and correct server issues. The system displays statuses (green, yellow, white, or red) for the Access Manager components. Health information can be accessed at the following places:

- *Access Manager > Overview*

  The Overview page shows the heath status at the component-level.
- *Access Manager > Auditing > Device Health*

  The Device Health page shows the health status for all devices in one list.
- *Access Manager > [Component] > Servers*

  The Servers page for each component provides a health status for each device.

This section discusses the following topics:

- Section 34.1, "Health States," on page 551
- Section 34.2, "Monitoring the Health of an Identity Server," on page 552
- Section 34.3, "Monitoring the Health of an Access Gateway," on page 553
- Section 34.4, "Viewing the Health of an Access Gateway Cluster," on page 557
- Section 34.5, "Monitoring the Health of an SSL VPN Server," on page 557

34.1 Health States

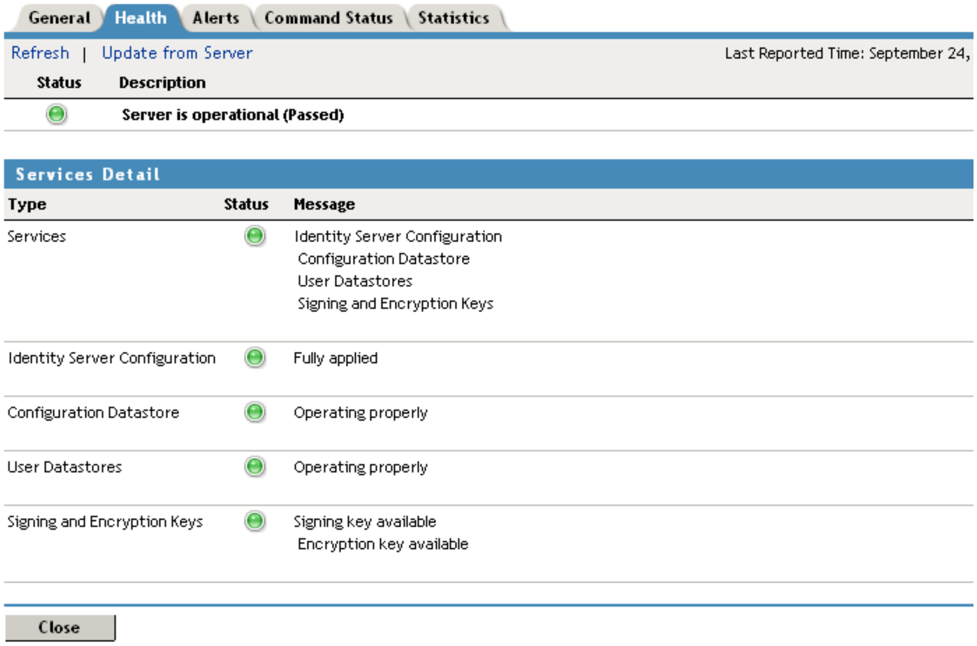
The Health page displays the current status of the server. The following states are possible:

Icon	**Description**
🟢	A green status indicates that the server has not detected any problems
⛔	A red status with a bar indicates that the server has been stopped.
○	A white status with disconnected bars indicates that the server is not communicating with the Administration Console.
◆	A yellow status indicates that the server might be functioning sub-optimally because of configuration discrepancies.
?	A yellow status with a question mark indicates that the server has not been configured.
✖	A red status with an x indicates that the server configuration might be incomplete or wrong, that a dependent service in not running or functional, or that the server is having a runtime problem.

34.2 Monitoring the Health of an Identity Server

To view detailed health status information for an Identity Server:

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *[Name of Server]* > *Health*.



Last Reported Time: September 24,

Status	Description
●	**Server is operational (Passed)**

Services Detail

Type	Status	Message
Services	●	Identity Server Configuration Configuration Datastore User Datastores Signing and Encryption Keys
Identity Server Configuration	●	Fully applied
Configuration Datastore	●	Operating properly
User Datastores	●	Operating properly
Signing and Encryption Keys	●	Signing key available Encryption key available

The status icon is followed by a description that explains the significance of the current state.

2 To ensure that the information is current, select one of the following:

- Click *Refresh* to refresh the page with the latest health available from the Administration Console.
- Click *Update from Server* to send a request to the Identity Server to update its status information. This can take a few minutes.

3 Examine the *Services Detail* section which displays the status of each service. For an Identity Server, this includes information such as the following:

Status Category	If not healthy
Status: Indicates whether the Identity Server is online and operational.	Verify whether the Identity Server has been stopped or is not configured. Also verify that network problems are not interfering with communications between the Identity Server and the Administration Console.
Services: Indicates the general health of all configured services.	If one service is unhealthy, this category reflects that status. See the particular service that also displays an unhealthy status.
Identity Server Configuration: Indicates the status of the configuration.	Configure the Identity Server or assign the server to a configuration. See Chapter 6, "Configuring an Identity Server," on page 55.

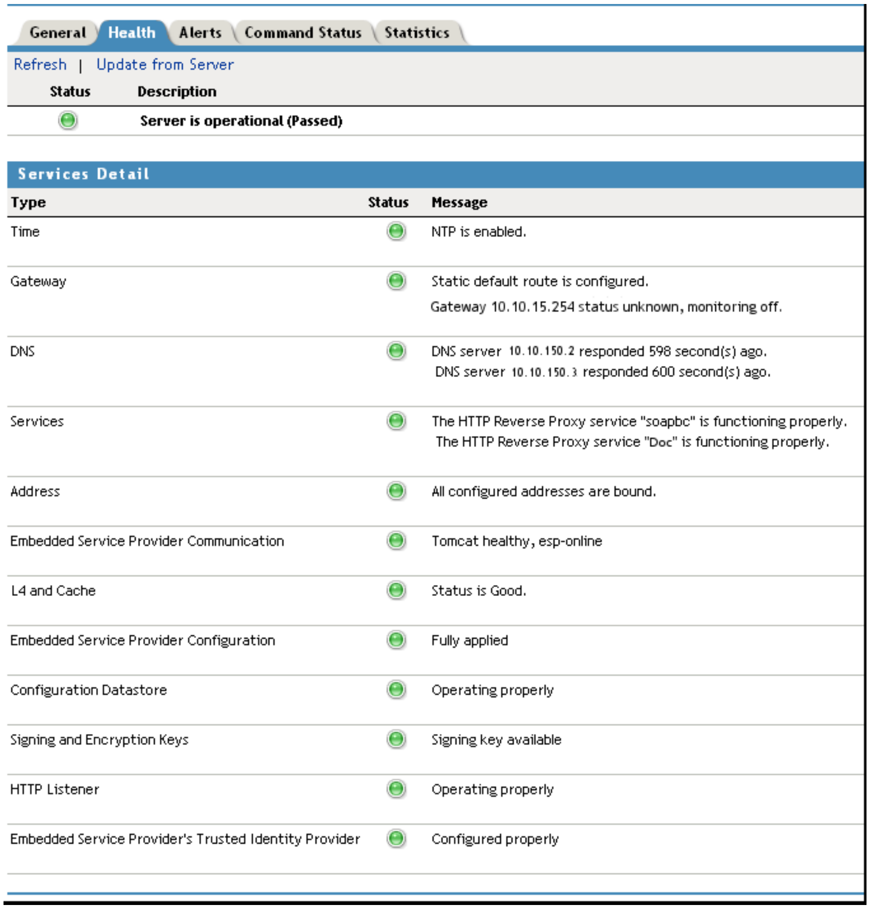
Status Category	If not healthy
Configuration Datastore: Indicates the status of the installed configuration datastore.	You might need to restart Tomcat or reinstall the Administration Console. If you have a backup Administration Console, you can restore it. See Section 2, "Backing Up and Restoring Components," on page 31. If you want to convert a secondary console to your primary console, see Section 37.5, "Converting a Secondary Console into a Primary Console," on page 580.
User Datastores: Indicates whether the Identity Server can communicate with the user stores, authenticate as the admin user, and find the search context.	Ensure that the user store is operating and configured correctly. You might need to import the SSL certificate for communication with the Identity Server. See Section 8.1, "Configuring Identity User Stores," on page 90.
Signing and Encryption Keys: Indicates the status of the signing and encryption keys for the Identity Server.	Renew or re-import the keys. See Section 6.5.3, "Managing the Keys, Certificates, and Trust Stores," on page 80.
SSL Communication: Indicates whether SSL communication is operating correctly. This health check appears only when the SSL communication check fails.	Check SSL connectivity. Check for expired SSL certificates.

4 Click *Close*.

34.3 Monitoring the Health of an Access Gateway

To view detailed health status information of an Access Gateway:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Server]* > *Health*.



Status	Description
●	**Server is operational (Passed)**

Services Detail

Type	Status	Message
Time	●	NTP is enabled.
Gateway	●	Static default route is configured. Gateway 10.10.15.254 status unknown, monitoring off.
DNS	●	DNS server 10.10.150.2 responded 598 second(s) ago. DNS server 10.10.150.3 responded 600 second(s) ago.
Services	●	The HTTP Reverse Proxy service "soapbc" is functioning properly. The HTTP Reverse Proxy service "Doc" is functioning properly.
Address	●	All configured addresses are bound.
Embedded Service Provider Communication	●	Tomcat healthy, esp-online
L4 and Cache	●	Status is Good.
Embedded Service Provider Configuration	●	Fully applied
Configuration Datastore	●	Operating properly
Signing and Encryption Keys	●	Signing key available
HTTP Listener	●	Operating properly
Embedded Service Provider's Trusted Identity Provider	●	Configured properly

The status icon is followed by a description that explains the significance of the current state.

2 To ensure that the information is current, select one of the following:

- Click *Refresh* to refresh the page with the latest health available from the Administration Console.
- Click *Update from Server* to send a request to the Access Gateway to update its status information. If you have made changes that affect the health of the Access Gateway, select this option. Otherwise, it can take up to five minutes for the health status to change.

3 Examine the *Services Detail* section which displays the status of each service. For an Access Gateway, this includes information such as the following:

Status Category	If not healthy
Status: Indicates whether the Access Gateway is online.	Check the status of the Enterprise Service Provider Configuration. If its status does not appear in the list of services, you need to start the service provider. In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Server]* > *Actions* > *Start Service Provider*. Also verify that network problems are not interfering with communications between the Access Gateway and the Administration Console.
Time: Indicates the type of time configuration. Time must be configured so that it remains synchronized with the other servers in the configuration (the Identity Server, SSL VPN server, J2EE agents, Web servers, etc.).	See Section 15.3, "Setting Date and Time," on page 253
Gateway: Specifies the type of routing that is configured for the gateway.	See Section 15.7.2, "Viewing and Modifying Gateway Settings," on page 266.
DNS: Specifies whether a domain name server has been configured and is active	(Linux only) Displays the IP address of the each configured DNS server and when the server last responded. (NetWare only) If the DNS server is configured but not configured for monitoring, the following message appears: `(Passed) Domain and DNS Servers configured`. If the DNS server is configured and monitoring is enabled, the following message appears: `(Passed) Domain and DNS Servers configured and active`. See Section 15.7.3, "Viewing and Modifying DNS Settings," on page 268.
Services: Indicates the general health of all configured services.	(Linux only) Displays messages about the health of the reverse proxy, the back-end Web servers, and internal services (the SOAP back channel and the communication module). (NetWare® only) Displays a general status message. For more information, see the particular services that also display an unhealthy status.
Address: Indicates whether an IP address has been configured for the reverse proxy to listen on. This is required for the Access Gateway to function.	See Section 13.1, "Creating a Reverse Proxy and Proxy Service," on page 200.
Reverse Proxy: Specifies whether a reverse proxy has been configured. An Access Gateway must have at least one reverse proxy configured.	See Section 13.1, "Creating a Reverse Proxy and Proxy Service," on page 200.

Status Category	If not healthy
Embedded Service Provider Communication: (Linux only) Indicates whether the embedded service provider can communicate with the Identity Server.	Restart the embedded service provider. If restarting the embedded service provider fails, try restarting Tomcat.

L4 and Cache: The L4 status indicates whether the Linux Access Gateway is responding to health checks from the L4 switch. The number increments with each health check for which the Access Gateway does not send a response.
- When it reaches 13, the health is changed to yellow.
- When it reaches 31, the health is changed to red.
If the Access Gateway recovers and starts responding, the health turns green after 20 seconds and the unresponsive count is reset to 0.
To fix the problem if it does not resolve itself, restart the Linux Access Gateway.
The cache status indicates the current number of delayed cache requests and whether enough memory is available to process new requests.
- When this number reaches 101, the health is changed to yellow.
- When this number reaches 151, the health changes to red. To solve the problem, you need to restart the Linux Access Gateway.
	Restart the Linux Access Gateway machine by entering the following commands:
	`/etc/init.d/novell-vmc stop`
	`/etc/init.d/novell-vmc start`

Embedded Service Provider Configuration: Specifies whether the Access Gateway has been configured to trust an Identity Server and whether that configuration has been applied. At least one Identity Server must be configured and set up as a trusted authentication source for the Access Gateway. A green status indicates that a configuration has been applied; it does not indicate that it is a functioning configuration.	See Chapter 6, “Configuring an Identity Server,” on page 55 for information on configuring an Identity Server. See Section 13.1, “Creating a Reverse Proxy and Proxy Service,” on page 200 for information on assigning an Identity Server configuration to the Access Gateway.
Configuration Data store: Indicates whether the configuration data store is functioning correctly.	See Section 2, “Backing Up and Restoring Components,” on page 31.
Signing and Encryption Keys: Indicates whether the Signing keystore contains a key.	Click *Access Gateways* > *Edit* > *Service Provider Certificates* > *Signing* and replace signing key in this keystore.
HTTP Listener: Indicates whether the Access Gateway and the embedded service provider are communicating.	Restart the Access Gateway. See Section 3.2.7, “Rebooting the Access Gateway,” on page 40.

Status Category	If not healthy
Embedded Service Provider’s Trusted Identity Provider: Indicates whether the configuration that the Access Gateway trusts has been configured to contain at least one Identity Server.	Modify the Identity Server configuration and add an Identity Server (see Section 6.1.2, “Assigning an Identity Server to a Cluster Configuration,” on page 60) or reconfigure the Access Gateway to trust a different Identity Server configuration (see Section 13.1, “Creating a Reverse Proxy and Proxy Service,” on page 200).

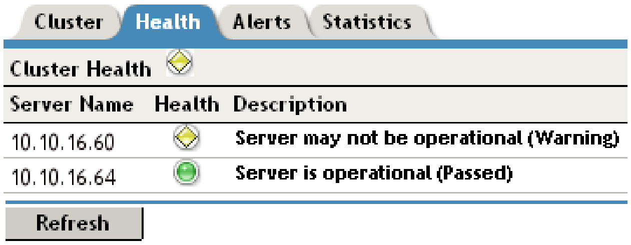
4 Click *Close*.

34.4 Viewing the Health of an Access Gateway Cluster

The *Health* icon on the cluster row displays the status of the least healthy member of the cluster. To view details about the status of the cluster:

1 In the Administration Console, click *Access Manager* > *Access Gateways*.

2 On the cluster row, click the *Health* icon.



Cluster Health ◆

Server Name	Health	Description
10.10.16.60	◆	Server may not be operational (Warning)
10.10.16.64	●	Server is operational (Passed)

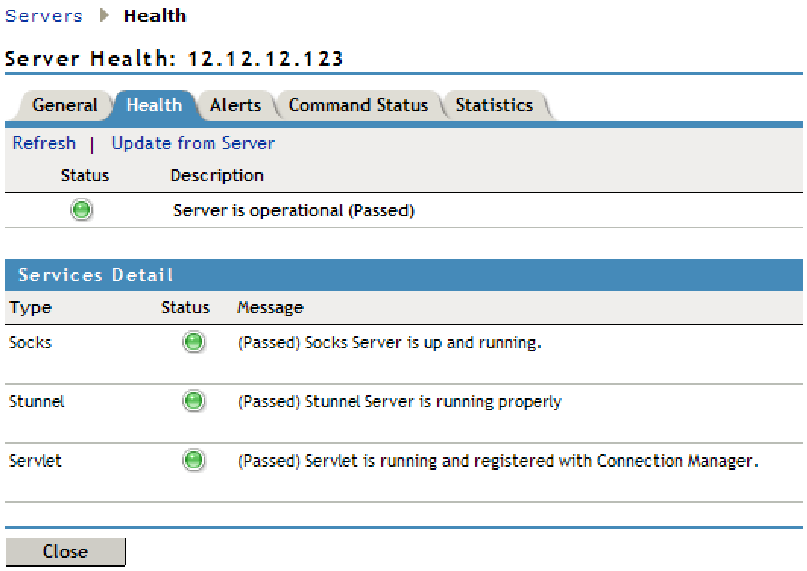
3 To ensure that the information is current, click *Refresh*.

4 To view specific information about the status of an Access Gateway, click the Health icon in the Access Gateway row. For more information, see Section 34.3, “Monitoring the Health of an Access Gateway,” on page 553.

34.5 Monitoring the Health of an SSL VPN Server

You can monitor the health of an SSL VPN Server through the Health page, which displays the current status of the server.

1 In Administration Console, click *Access Manager* > *SSL VPNs* > *[Server Name]* > *Alerts*.



Server Health: 12.12.12.123

Status	Description
●	Server is operational (Passed)

Services Detail

Type	Status	Message
Socks	●	(Passed) Socks Server is up and running.
Stunnel	●	(Passed) Stunnel Server is running properly
Servlet	●	(Passed) Servlet is running and registered with Connection Manager.

The *Status* column displays the current state, and the *Description* column explains the significance of the current state.

The *Services Details* section provides the following information:

Type: Specifies the type of service.

Status: Specifies the status of the service.

Message: Specifies a description of the status of the service.

2. To reload the current page with the latest status, click *Refresh*.
3. To send a request to the agent to update its status information, click *Update from Server*. Click *OK* in the confirmation dialog box. This can take a few minutes.
4. To close the Health page, click *Close*.

Reviewing Command Status

35

Commands are issued to a device when you make configuration changes and when you select an action such as stopping or starting a device.

Certain commands, such as start and stop commands, retry up to 10 times before they fail. The first few retries are spaced a couple minutes apart, then they move to 10 minute intervals. Such a command can take over an hour for it to result in a failure. As long as the command is in the retry cycle, the command has a status of pending.

- If you do not want to wait for the cycle to complete, you need to manually delete the command.
- If you enter the same command and it succeeds before the first command has completed its retry cycle, the first command will always stay in the pending state. You need to manually delete the command.

To view detailed information about the command status of a device, see one of the following sections:

- Section 35.1, “Viewing the Command Status of the Identity Server,” on page 559
- Section 35.2, “Viewing the Command Status of the Access Gateway,” on page 560
- Section 35.3, “Viewing Command Status of the SSL VPN Server,” on page 561

35.1 Viewing the Command Status of the Identity Server

The Command Status page lists scheduled events and the current status of each event. A new command appears in the list each time you change a configuration. The commands remain listed until you delete them.

1. In the Administration Console, click *Access Manager* > *Identity Servers*.
2. Click the *Command Status* link for the server.
3. To delete a command, select it and click *Delete*.
4. Click *Refresh* to refresh the display.

The following table describes the columns on the Command Status page:

Column Name	Description
Name	Lists the Identity Server name.
Status	Lists the status of each server.
Type	Displays type of command issued to the server.
Admin	Displays the credentials of the administrator who performed the command.
Date & Time	The date and time that the command was issued. Date and time entries are specified in the local time.

35.2 Viewing the Command Status of the Access Gateway

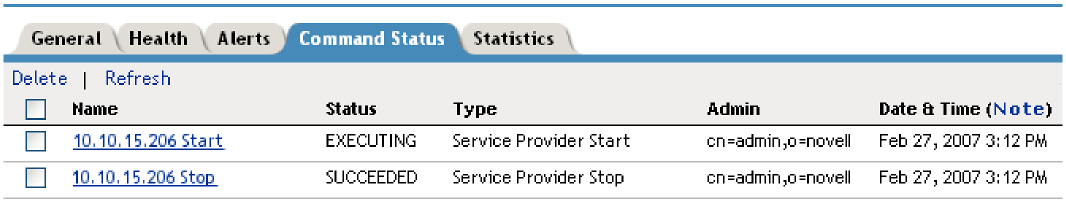
You can view the status of the commands that have been sent to the Access Gateway for execution. The *Apply Changes* button on the configuration page issue a command, and the results appear on this page. The Actions options, such as restarting the embedded service provider or purging the cache, also appear on this page.

This section describes the following tasks related to commands:

- Section 35.2.1, “Viewing the Status of Current Commands,” on page 560
- Section 35.2.2, “Viewing Detailed Command Information,” on page 561

35.2.1 Viewing the Status of Current Commands

1 In the Administration Console, click *Access Manager > Access Gateways > [Name of Server] > Command Status*.



☐	Name	Status	Type	Admin	Date & Time (Note)
☐	10.10.15.206 Start	EXECUTING	Service Provider Start	cn=admin,o=novell	Feb 27, 2007 3:12 PM
☐	10.10.15.206 Stop	SUCCEEDED	Service Provider Stop	cn=admin,o=novell	Feb 27, 2007 3:12 PM

This page lists the current commands and the following information about the commands:

Column Name	Description
Name	Contains the display name of the command. Click the link to view additional details about the command. For more information, see Section 35.2.2, “Viewing Detailed Command Information,” on page 561.
Status	Specifies the status of the command. Some of the possible states of the command include Pending, Incomplete, Executing, and Succeeded.
Type	Specifies the type of command.
Admin	Specifies if the system or a user issued the command. If a user issued the command, the DN of the user is displayed.
Date & Time	Specifies the local date and time the command was issued.

2 Select one of the following actions:

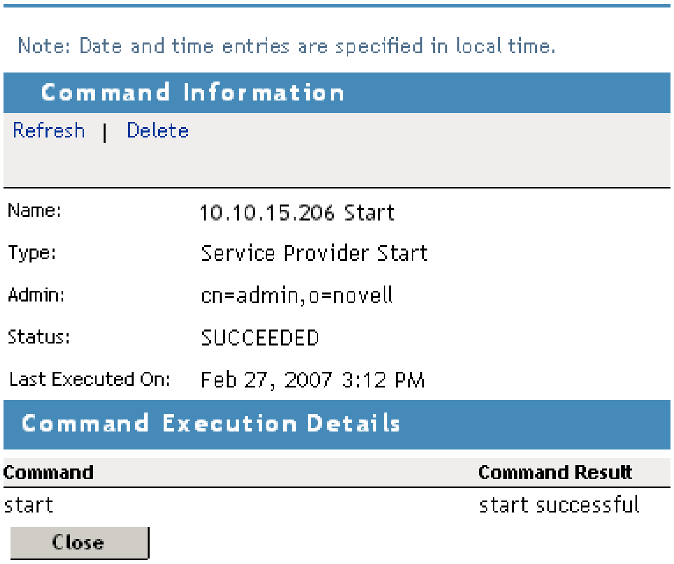
- To view information about a particular command, click the name of a command.
- To delete a command from the list, select the command, then click *Delete*.
- To refresh the status of the listed commands, click *Refresh*.

3 Click *Close*.

35.2.2 Viewing Detailed Command Information

To view information about an individual command:

1. In Administration Console, click *Access Manager > Access Gateways > [Name of Server] > Command Status.*
2. Click the name of a command to get detailed information.



Note: Date and time entries are specified in local time.

Name:	10.10.15.206 Start
Type:	Service Provider Start
Admin:	cn=admin,o=novell
Status:	SUCCEEDED
Last Executed On:	Feb 27, 2007 3:12 PM

Command	Command Result
start	start successful

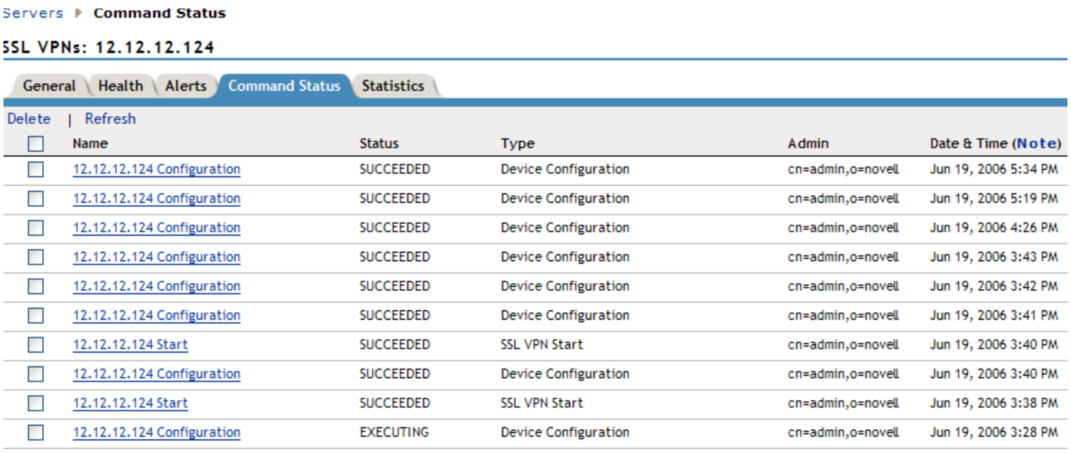
   To determine if any problems occurred, view the *Command Execution Details* section.

3. Select one of the following actions:
   - **Delete:** To delete a command, click *Delete*. Click *OK* in the confirmation dialog box.
   - **Refresh:** To update the current cache of recently executed commands, click *Refresh*.
4. Click *Close* to return to the Command Status page.

35.3 Viewing Command Status of the SSL VPN Server

Use the Command Status page to view the command status of the selected SSL VPN server.

1. In Administration Console, click *Access Manager > SSL VPNs > [Server Name] > Command Status*.



☐	Name	Status	Type	Admin	Date & Time (Note)
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 5:34 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 5:19 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 4:26 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 3:43 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 3:42 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 3:41 PM
☐	12.12.12.124 Start	SUCCEEDED	SSL VPN Start	cn=admin,o=novell	Jun 19, 2006 3:40 PM
☐	12.12.12.124 Configuration	SUCCEEDED	Device Configuration	cn=admin,o=novell	Jun 19, 2006 3:40 PM
☐	12.12.12.124 Start	SUCCEEDED	SSL VPN Start	cn=admin,o=novell	Jun 19, 2006 3:38 PM
☐	12.12.12.124 Configuration	EXECUTING	Device Configuration	cn=admin,o=novell	Jun 19, 2006 3:28 PM

This page lists the command and the following information about the command:

- **Name:** Contains the display name of the command. Click the link to view additional details about the command. For more information, see Section 35.3.1, "Viewing Command Information," on page 562.
- **Status:** Specifies the status of the command. Some of the possible states of the command include *Pending*, *Incomplete*, *Executing*, and *Succeeded*.
- **Type:** Specifies the type of command.
- **Admin:** Specifies if the system or a user issued the command. If a user issued the command, the DN of the user is displayed.
- **Date & Time:** Specifies the local date and time the command was issued.

2 To delete a command, select the check box for the command, then click *Delete*. The selected command is cleared.

3 To update the current cache of recently executed commands, click *Refresh*.

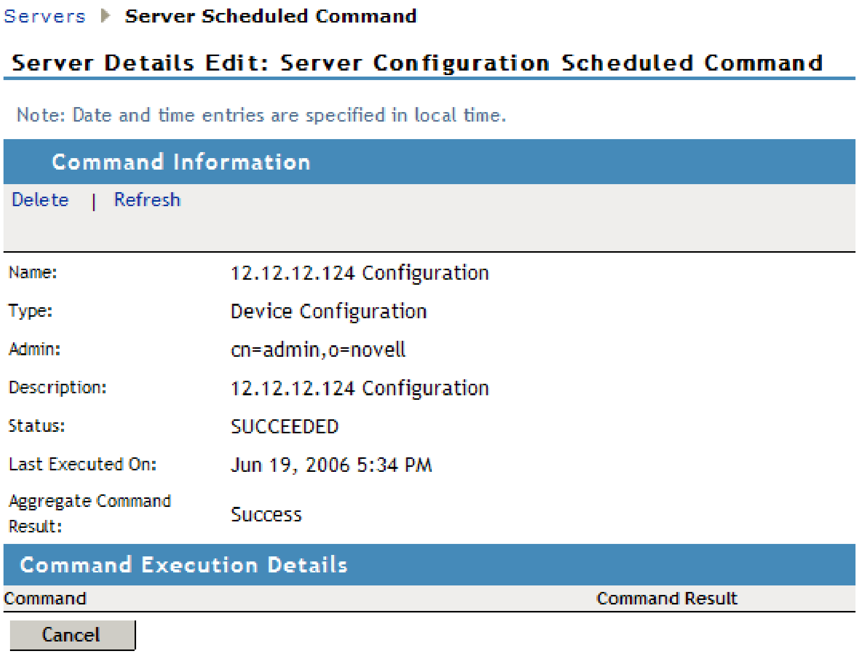
4 Click *Close* to close the Command Status page.

35.3.1 Viewing Command Information

To view configuration of individual commands:

1 In Administration Console, click *Access Manager > SSL VPNs > [Server Name] > Command Status >[Individual Command]*.The command status page is displayed.

2 Click the command to get a detailed information on the command. The Server Configuration scheduled command page is displayed.


Servers ▸ Server Scheduled Command
Server Details Edit: Server Configuration Scheduled Command
Command Information
Delete | Refresh
Command Execution Details
Command
Command Result
Cancel

Note: Date and time entries are specified in local time.

Name:	12.12.12.124 Configuration
Type:	Device Configuration
Admin:	cn=admin,o=novell
Description:	12.12.12.124 Configuration
Status:	SUCCEEDED
Last Executed On:	Jun 19, 2006 5:34 PM
Aggregate Command Result:	Success

You can perform the following actions:

- **Delete:** To delete a command, click *Delete*. Click *OK* in the confirmation dialog box.
- **Refresh:** To update the current cache of recently executed commands, click *Refresh*.

3 Click *Close* to return to the command status page.

Reviewing Alerts

36

36.1 Monitoring Identity Server Alerts

The Alerts page allows you to view information about current Java alerts and to clear them. An alert is generated whenever the Identity Server detects a condition that prevents it from performing normal system services.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *[Name of Server]* > *Alerts* tab.

2 To acknowledge an alert, select the check box for the alert, then click *Acknowledge Alert(s)*. When you acknowledge an alert, you clear the alert from the list.

3 Click *Close*.

4 (Optional) To verify that the problem has been solved, *Identity Servers* > *[Name of Server]* > *Health* > *Update from Server*.

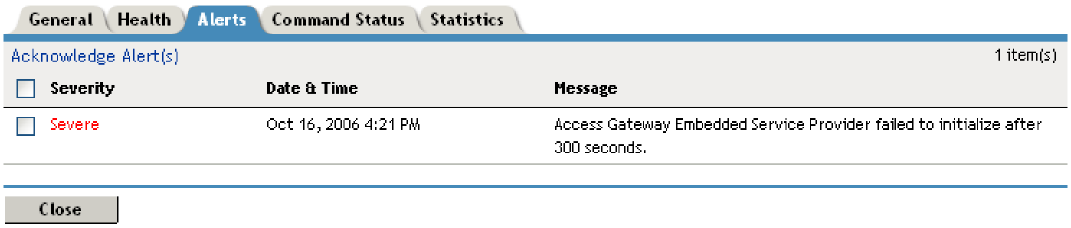
36.2 Monitoring Access Gateway Alerts

The Access Gateway has been programmed to issue events to various types of systems (such as a Novell® Audit server or a Syslog server) so that the administrator can be informed when significant changes occur that modify how the Access Gateway is performing. For information about auditing and audit events, see Chapter 31, “Enabling Auditing,” on page 505. This section describes how to use the following types of alerts:

36.2.1 Reviewing Java Alerts

The Alerts page allows you to view information about current Java alerts and to clear them. An alert is generated whenever the Access Gateway detects a condition that prevents it from performing normal system services.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Server]* > *Alerts*.



☐ Severe | Oct 16, 2006 4:21 PM | Access Gateway Embedded Service Provider failed to initialize after 300 seconds.

2. To acknowledge an alert, select the check box for the alert, then click *Acknowledge Alert(s)*. When you acknowledge an alert, you clear the alert from the list.
3. Click *Close*.
4. (Optional) To verify that the problem has been solved, click *Access Gateways > [Server Name] > Health > Update from Server*.

The NetWare® Access Gateway currently sends the following severe alerts when it is not functioning correctly:

Alert Message	Solution

`Access Gateway Embedded Service Provider failed to initialize`
`Access Gateway Server communication channel failed to start`
	Click *Access Gateways,* select the Access Gateway, then click *Actions > Service Provider > Start Service Provider.*

36.2.2 Configuring Access Gateway Alerts

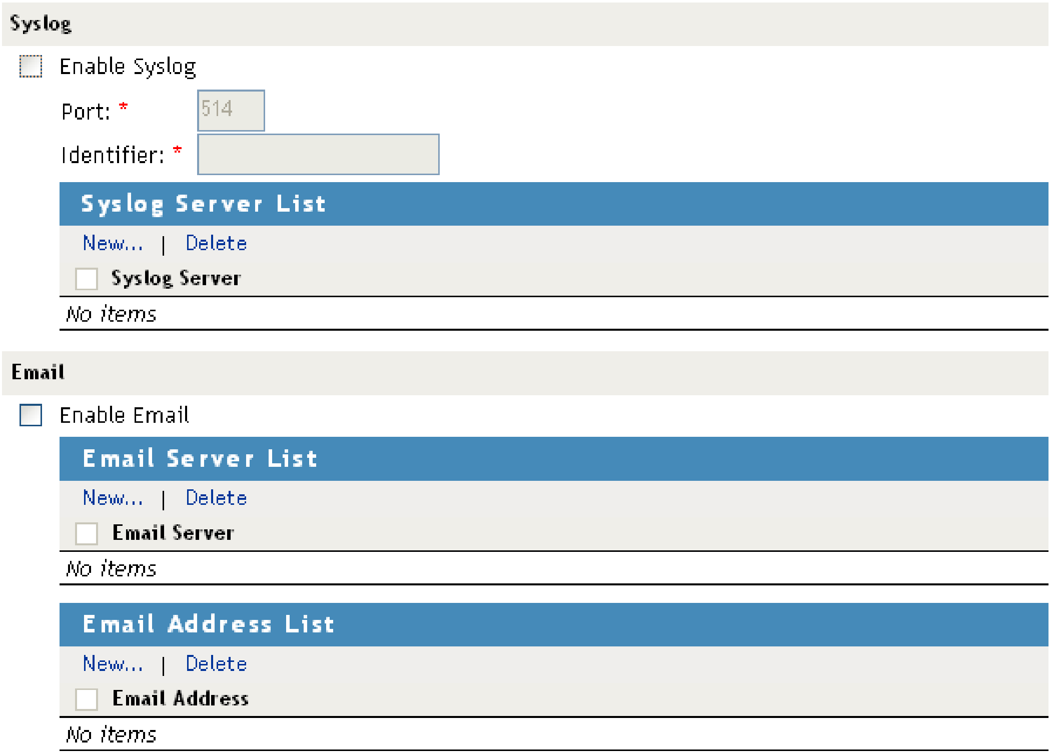
The configuration steps for Access Gateway Alerts are platform-specific, although both platforms support similar options. To set up notification for these types of alerts, see the following sections:

- "NetWare Access Gateway Alerts" on page 566
- "Linux Access Gateway Alerts" on page 568
- "Access Gateway Cluster Alerts" on page 571

NetWare Access Gateway Alerts

For a NetWare® Access Gateway, the *Legacy Alerts* option allows you to send notification of generated system alerts to a Syslog server, to a list of e-mail recipients, or to both.

1. In the Administration Console, click *Access Manager > Access Gateways > Edit > Legacy Alerts*.



Syslog

☐ Enable Syslog

Port: * 514

Identifier: *

Syslog Server List

New... | Delete

☐ **Syslog Server**

No items

Email

☐ Enable Email

Email Server List

New... | Delete

☐ **Email Server**

No items

Email Address List

New... | Delete

☐ **Email Address**

No items

2 Enable the Syslog services by configuring the following fields:

Enable Syslog: Selecting this option enables syslog alerts. You must also configure a Syslog server and select some alerts. (See Step 3 and Step 5.)

Port: Specifies the port where the Syslog server listens for Syslog messages. The default value for the UDP port is 514. Make sure this port value matches the port configuration of your Syslog server.

Identifier: Specifies a string that identifies the Access Gateway as the generator of the alert.

3 If you enabled Syslog services, configure a Syslog server.

3a Click *New* under the *Syslog Server List*.

3b Specify the DNS name or IP address of the Syslog server and click *OK*.

4 To enable e-mail notification, select *Enable Email*.

You must also set up an e-mail server and a list of recipients and select some alerts before any alert notifications are sent

4a Click *New* under the *Email Server List* section and specify the DNS name or IP address of your e-mail server.

Repeat this step if you have more than one e-mail server.

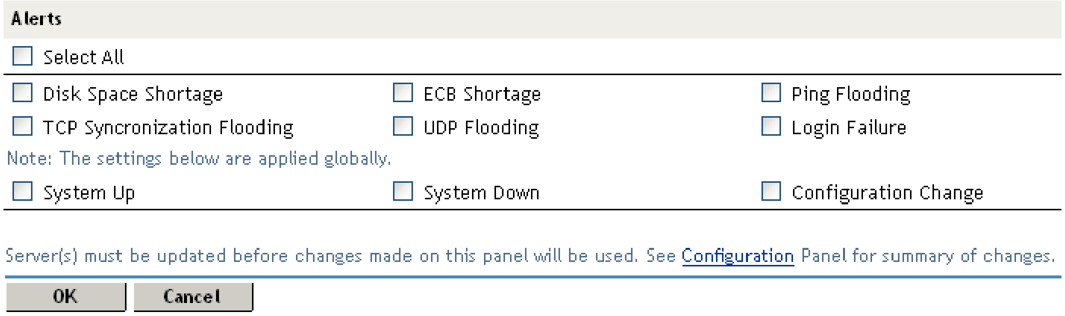
4b Click *Email Address* to activate all servers in the list, or click the box by individual servers to select only some servers in the list.

4c Click *New* under the *Email Address List* section and specify the e-mail address of the user you want to receive alert notifications.

Repeat this step to add others to the list.

4d Click *Email Address* to activate notification for all users in the list, or click the box by individual users to select only some users in the list.

5 Select the alerts for notification.



Alerts

☐ Select All

☐ Disk Space Shortage ☐ ECB Shortage ☐ Ping Flooding

☐ TCP Syncronization Flooding ☐ UDP Flooding ☐ Login Failure

Note: The settings below are applied globally.

☐ System Up ☐ System Down ☐ Configuration Change

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

Select All: Select this option for all alerts. Otherwise, select one or more of the following:

Alert	Description
Disk Space Shortage	Generated when disk space is low on the OS (sys:) or Log (log:) volumes.
TCP Synchronization Flooding	Generated when TCP/IP detects a flooding of synchronization packets. This often happens during a denial-of-service attack.
ECB Shortage	Generated when network receive buffers are low.
UDP Flooding	Generated when TCP/IP detects a flooding of UDP packets. This often happens during a denial-of-service attack.
Ping Flooding	Generated when TCP/IP detects a flooding of ping packets. This often happens during a denial-of-service attack.
Login Failure	Generated each time a login failure occurs from the management tool or from FTP. The alert contains the IP address of the client making the unsuccessful attempt.
System Up	Generated each time the Access Gateway is started.
System Down	Generated each time the Access Gateway is stopped.
Configuration Change	Generated each time the configuration of the Access Gateway is modified.

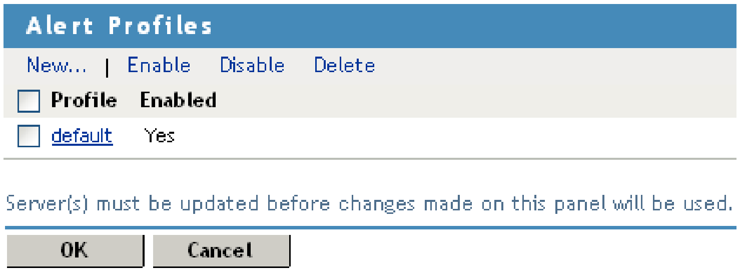
6 To save your modifications, click *OK* twice.

7 On the Access Gateways page, click *Update*.

Linux Access Gateway Alerts

For a Linux Access Gateway, this option allows you to send notification of generated system alerts to a Syslog server, to SNMP, to a system controller, to a log file, or to a list of e-mail recipients.

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Alerts*.


Alert Profiles

New... | Enable Disable Delete

☐ Profile	Enabled
☐ default	Yes

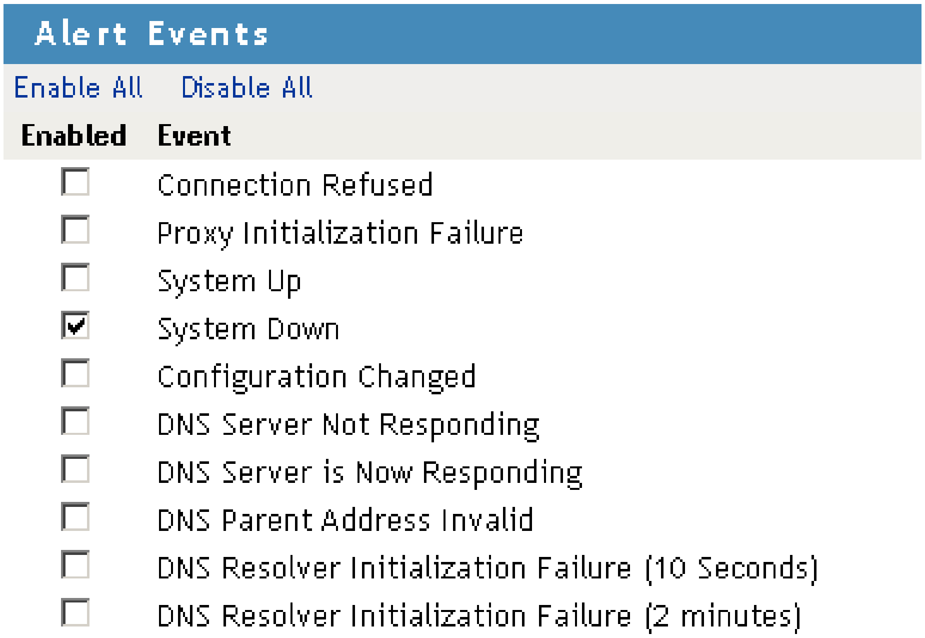
Server(s) must be updated before changes made on this panel will be used.

OK Cancel

2 To add a new profile, click *New*.

3 Specify a name for the profile, then click *OK*.

4 Click the new profile to configure alert events.


Alert Events

Enable All Disable All

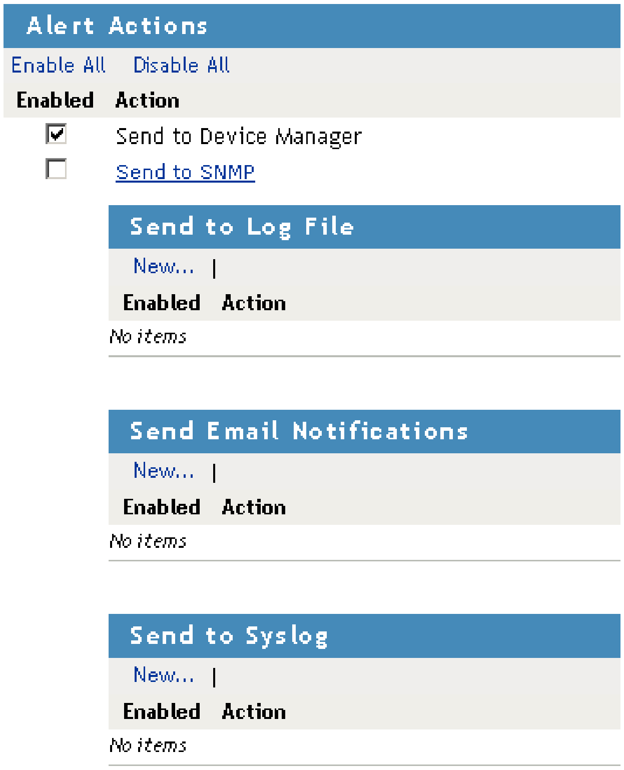
Enabled	Event
☐	Connection Refused
☐	Proxy Initialization Failure
☐	System Up
☑	System Down
☐	Configuration Changed
☐	DNS Server Not Responding
☐	DNS Server is Now Responding
☐	DNS Parent Address Invalid
☐	DNS Resolver Initialization Failure (10 Seconds)
☐	DNS Resolver Initialization Failure (2 minutes)

5 To select the alerts for notification, select one or more of the following:

Alert	Description
Connection Refused	Generated when the connection is refused.
Proxy Initialization Failure	Generated when the embedded service provider fails to initialize.
System Up	Generated each time the Access Gateway is started.
System Down	Generated each time the Access Gateway is stopped.
Configuration Changed	Generated each time the configuration of the Access Gateway is modified.
DNS Server Not Responding	Generated each time the DNS server fails to respond.
DNS Server Is Now Responding	Generated each time the DNS server comes up.

Alert	Description
DNS Parent Address Invalid	Generated when the IP address of DNS parent is invalid.
DNS Resolver Initialization Failure (10 seconds)	Generated when the DNS resolver initialization fails.
DNS Resolver Initialization Failure (2 minutes)	Generated when the DNS resolver initialization fails.

6 To send alerts to all destinations, click *Enable All*. Otherwise, select the action for each destination.



☑ Send to Device Manager

☐ Send to SNMP

No items

No items

No items

7 To send alerts to the Administration Console select the *Send to Device Manager* check box.

8 To send alerts to an SNMP server, select the *Send to SNMP* check box, then click the *Send to SNMP* link.

8a Specify the IP address of the SNMP server in the *IP Addresse*s field, then click *Insert*.

You can add multiple IP addresses in the same text box if you separate them with a comma, or you can add them individually one after the other.

8b To delete an IP address, click the *Delete* button next to the IP address that you want to delete. Click *OK* in the confirmation dialog box.

9 To send alerts to a log file, click *New*, then specify a name for the log profile.

9a Configure the following Log File details:

- **Log File Name:** Specify a name for the log file and a path where the file should be stored.

- **Max File Size:** Specify a maximum size in KB for the log file. The size can be from 50 to 100000 KB. Specify 0 to indicate that there is no maximum file size.

9b Click *OK*.

10 To enable e-mail notification click *New*, then specify a name for the e-mail profile.

10a Configure the following e-mail details:

- **E-mail Recipients:** Specify the e-mail address of the recipient, then click *Insert*. You can add multiple e-mail addresses. Click *Delete* to delete any of the e-mail addresses, then click *OK* in the confirmation dialog box.
- **Mail Exchange Servers:** Specify the IP address or the DNS name of the mail exchange server. Click *Delete* to delete any of the mail exchange servers addresses, then click *OK* in the confirmation dialog box.

10b Click *OK*.

11 To enable syslog alerts click *New*, then specify a name for the Syslog profile.

11a Configure the following syslog details:

- **Facility Name:** Specify a facility name for the Syslog server. It can be any name from local0 to local7. If you specify local0 as your facility name, the alerts are stored at `\var\logs\ics_dyn.log`. The Linux Access Gateway uses local0 for normal logging information. Therefore, it is not recommended to specify local0 as your facility name.

11b Click *OK*.

11c To delete a syslog profile, click *Delete*. Click *OK* in the confirmation dialog box.

12 To delete an Alert Profile, select the profile, then click *Delete*. Click *OK* in the confirmation dialog box.

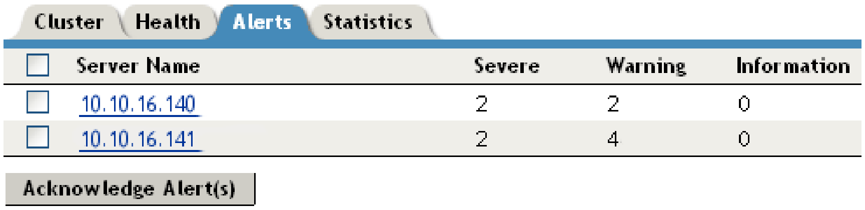
13 To save your modifications, click *OK* twice.

14 On the *Access Gateways* page, click *Update*.

Access Gateway Cluster Alerts

To view information about current alerts for all members of a cluster:

1 In the Administration Console, click *Access Manager* > *Access Gateways* > *[Name of Cluster]* > *Alerts*.



☐	Server Name	Severe	Warning	Information
☐	10.10.16.140	2	2	0
☐	10.10.16.141	2	4	0

2 Analyze the data displayed in the table.

Column	Description
Server Name	Lists the name of the Access Gateway that sent the alert. To view additional information about the alerts for a specific Access Gateway, click the name of an Access Gateway.
Severe	Lists the number of critical alerts that have been sent and not acknowledged.
Warning	Lists the number of warning alerts that have been sent and not acknowledged.
Information	Lists the number of informational alerts that have been sent and not acknowledged.

3. To acknowledge all alerts for an Access Gateway, select the check box for the Access Gateway, then click *Acknowledge Alert(s)*. When you acknowledge an alert, you clear the alert from the list.
4. To view information about a particular alert, click the server name. For information about a specific alert, see Section 36.2.1, "Reviewing Java Alerts," on page 565.

36.2.3 Enabling SNMP

(NetWare only) The SNMP page allows you configure the Access Gateway with basic SNMP information so the Access Gateway can communicate with your SNMP management workstations.

This SNMP implementation follows the ISO SNMP version 1 standard outlined in RFC 1067: A Simple Network Management Protocol (http://www.faqs.org/rfcs/rfc1067.html).

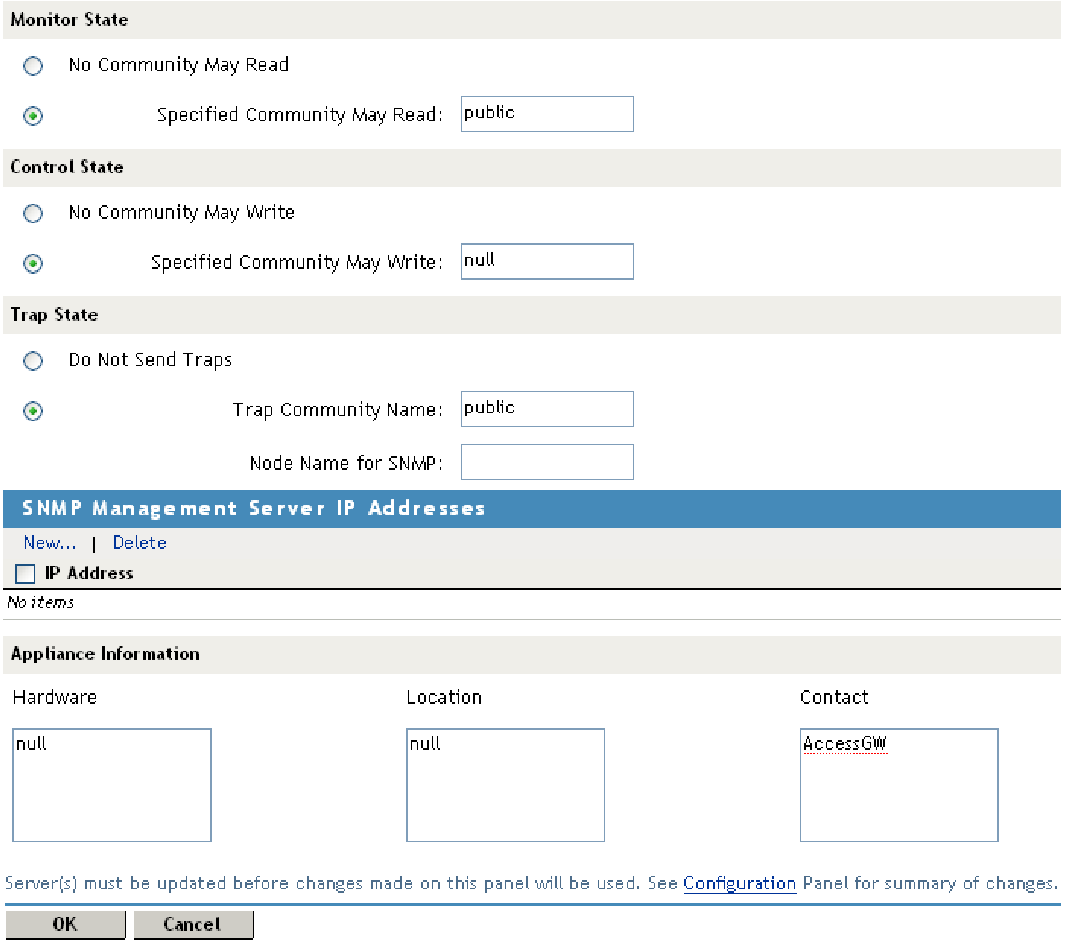
When SNMP-enabled components of Access Gateway start, they register with the system. When the system receives a request for a specific SNMP parameter, it knows which component to contact to obtain the information.

The Access Gateway has an `ichain.mib` file in the `sys:\etc\proxy\data` directory. To see a list of standard SNMP parameters, use the FTP get command to retrieve this file, then compile it for use with your SNMP management software.

If you specify a trap community name and specify an SNMP management workstation on the SNMP page, all alerts you select in the Legacy Alerts page (see "NetWare Access Gateway Alerts" on page 566) are automatically sent as SNMP traps even if you have not configured syslog or e-mail alert notification on the Legacy Alerts page.

To set up SNMP:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *SNMP*.



Monitor State

○ No Community May Read

● Specified Community May Read: public

Control State

○ No Community May Write

● Specified Community May Write: null

Trap State

○ Do Not Send Traps

● Trap Community Name: public

Node Name for SNMP:

SNMP Management Server IP Addresses

New... | Delete

☐ IP Address

No items

Appliance Information

Hardware	Location	Contact
null	null	AccessGW

Server(s) must be updated before changes made on this panel will be used. See Configuration Panel for summary of changes.

2 Configure the following:

Monitor State: Specifies whether the community has Read access to monitor the Access Gateway. If it does, you need to specify the community name. Community names must contain only ASCII characters and must not have spaces.

Control State: Specifies whether the community has Write access to the control states of the Access Gateway. If it does, you need to specify the community name. Community names must contain only ASCII characters and must not have spaces.

Trap State: Specifies whether traps are sent. If they are sent, you can specify a community (location, IP octets, or other identifier) from which traps are sent to the management stations you designate. Community names must contain only ASCII characters and must not have spaces. You can also specify a *Node Name for SNMP* for management of the Access Gateway through SNMP.

3 Add an SNMP server.

3a In the *SNMP Management Server IP Addresses* section, click *New*.

3b Specify the IP address of the SNMP server, then click *OK*.

3c Repeat to add additional servers.

4 (Optional) Configure appliance information.

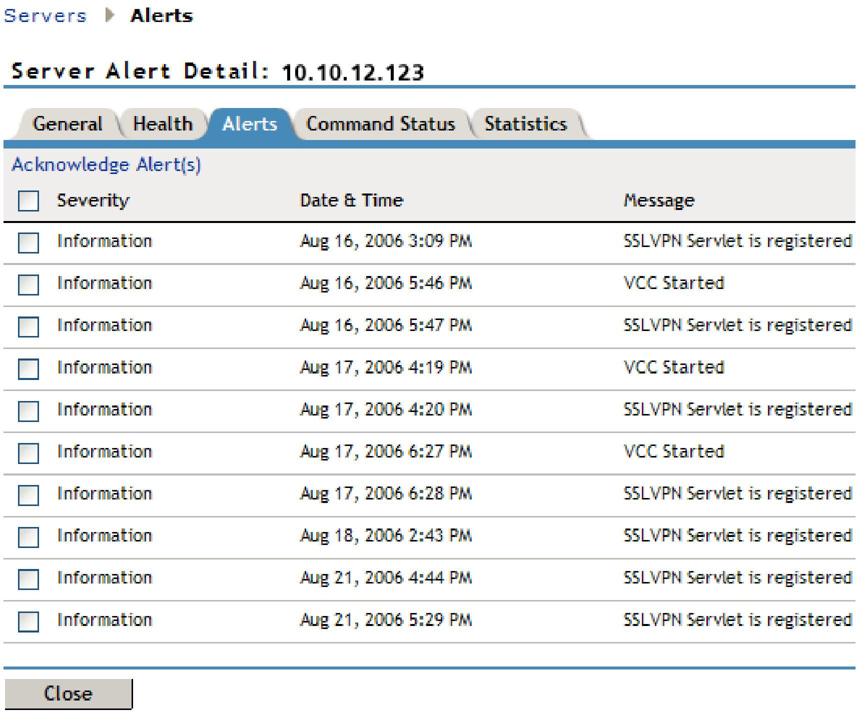
The *Appliance Information* fields allow you to enter additional information about the Access Gateway. You can describe the Access Gateway hardware and its location, and provide the name of the person responsible for the Access Gateway.

5 To save your modifications, click *OK* twice, then on the Access Gateways page, click *Update*.

36.3 Monitoring SSL VPN Alerts

The *Alerts* page allows you to view information about current system alerts and to clear them. An alert is generated whenever the SSL VPN Gateway detects a condition that prevents it from performing normal system services.

1 In Administration Console, click *Access Manager > SSL VPNs > [Server Name] > Health*.



☐	Severity	Date & Time	Message
☐	Information	Aug 16, 2006 3:09 PM	SSLVPN Servlet is registered
☐	Information	Aug 16, 2006 5:46 PM	VCC Started
☐	Information	Aug 16, 2006 5:47 PM	SSLVPN Servlet is registered
☐	Information	Aug 17, 2006 4:19 PM	VCC Started
☐	Information	Aug 17, 2006 4:20 PM	SSLVPN Servlet is registered
☐	Information	Aug 17, 2006 6:27 PM	VCC Started
☐	Information	Aug 17, 2006 6:28 PM	SSLVPN Servlet is registered
☐	Information	Aug 18, 2006 2:43 PM	SSLVPN Servlet is registered
☐	Information	Aug 21, 2006 4:44 PM	SSLVPN Servlet is registered
☐	Information	Aug 21, 2006 5:29 PM	SSLVPN Servlet is registered

The following information is displayed:

- **Severity:** Describes the type of alert. An alert can be informational, critical, or a warning.
- **Date & Time:** Indicates the date and time when an alert was issued. The date and time are given in the local time.
- **Message:** Displays the message that was sent with the alert. This information is optional.

2 To send an acknowledgement, select the check box next to the alert, then click *Acknowledge Alert(s)*. When you acknowledge an alert, the alert is cleared from the list.

3 Click *Close* to close the Alerts page.

Troubleshooting

VIII

The following sections contain information about troubleshooting the components of Access Manager:

For a description of the event codes, see *Novell Access Manager 3.0 SP3 Event Codes*.

Troubleshooting the Administration Console

37

This section discusses general troubleshooting issues found in the Administration Console:

- Section 37.1, “Checking for Potential Configuration Problems,” on page 577
- Section 37.2, “Logging,” on page 579
- Section 37.3, “Event Codes,” on page 579
- Section 37.4, “Fixing a Failed Secondary Console,” on page 579
- Section 37.5, “Converting a Secondary Console into a Primary Console,” on page 580
- Section 37.6, “Session Conflicts,” on page 584
- Section 37.7, “Unable to Log In to the Administration Console,” on page 585
- Section 37.8, “Exception Processing IdentityService_ServerPage.JSP,” on page 585
- Section 37.9, “Backup/Restore Failure Because of Special Characters in Passwords,” on page 585

37.1 Checking for Potential Configuration Problems

If your Access Manager components are not running as you have configured them to run, you might want to check the system to see if any of the components have configuration or network problems.

1. In the Administration Console, click *Access Manager* > *Auditing* > *Troubleshooting* > *Configuration*.
2. All of the options should be empty, except the *Cached Access Gateway Configurations* option (see Step 4). If an option contains an entry, you need to clear it. Select the appropriate action from the following table:

Option	Description and Action
Device Pending with No Commands	If you have a device that remains in the pending state, even when all commands have successfully executed, that device appears in this list. Before deleting the device from this list, check its Command Status. If the device has any commands listed, select them, then delete them. Wait a few minutes. If the device remains in a pending state, return to this troubleshooting page. Find the device in the list, then click *Remove*. The Administration Console clears the pending state.

Option	Description and Action
Other Known Device Manager Servers	If a secondary Administration Console is in a non-reporting state, perhaps caused by hardware failure, its configuration needs to be removed from the primary Administration Console. As long as it is part of the configuration, other Access Manager devices try to contact it. If you cannot remove it by running the uninstall script on the secondary Administration Console, you can remove it by using this troubleshooting page. Click the *Remove* button next to the console that is in the non-reporting state. All references to the secondary Administration Console are removed from the configuration database.
Access Gateways with Incomplete Proxy Configuration	If you start to configure a reverse proxy, but you fail to complete the process by configuring a proxy service and selecting an IP address and port, the file used to update the Access Gateway contains an invalid configuration. You can return to the Access Gateway, and either delete the partial configuration or complete it. These actions create a valid configuration that can then be used to update the server. Or, click the *Remove* button next to the proxy that has an incomplete configuration. This removes the invalid reverse proxy configuration.
Access Gateways with Corrupt Protected Resource Data	If you modify the configuration for a protected resource, update the Access Gateway with the changes, then review the configuration for the protected resource and the changes have not been applied, the configuration for the protected resource is corrupted. Click the *Repair* button next to the protected resource that has a corrupted configuration. You should then be able to modify its configuration, and when you update the Access Gateway, the changes should be applied and saved.
Access Gateways with Duplicate Protected Resource Data	After an upgrade, if you get errors related to invalid content for policy enforcement lists, you need to correct them. The invalid elements that do not have an associated resource data element are listed in this section. Click the *Repair* button to remove them.
Access Gateways with Protected Resources Referencing Nonexistent Policies	Protected resources have problems when policies are deleted before their references to the protected resources are removed. If you have protected resources in this condition, they are listed in this section. Click the *Repair* button to remove these references. Then verify that your protected resources have the correct policies enabled. Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Protected Resources*, then change to the *Policy View*.

Option	Description and Action
Access Gateways with Invalid Alert Profile References	You can create XML validation errors on your Linux Access Gateway if you start to create an alert profile (click *Access Gateways* > *Edit* > *Alerts* > *New*), but you do not finish the process. The incomplete alert profile does not appear in the configuration for the Access Gateway, so you cannot delete it. If such a profile exists, it appears in the *Access Gateways with Invalid Alert Profile References* list. Click the *Remove* button by the invalid profile. You should then be able to modify its configuration, and when you update the Access Gateway, the changes should be applied and saved.

3. When you have finished repairing or deleting invalid Access Gateway configurations, click the *Access Gateways* link, then click *Update* > *OK*.
4. (Optional) To verify that all members of an Access Gateway cluster have the same configuration in cache, click *Auditing* > *Troubleshooting* > *Configuration*.
5. Scroll to the *Cached Access Gateway Configuration* option, then click *View* next to the cluster configuration or next to an individual Access Gateway.

   This option allows you to view the Access Gateway configuration that is currently residing in browser cache. If the Access Gateway belongs to a cluster, you can view the cached configuration for the cluster as well as the cached configuration for each member. The + and - buttons allow you to expand and collapse individual configurations. The configuration is displayed in XML format

   To search for particular configuration parameters, you need to copy and paste the text into a text editor.

37.2 Logging

You can troubleshoot by configuring component logging. In the Administration Console, click *Identity Server* > *Servers* > *Edit* > *Logging*.

See Section 32.2, “Configuring Identity Server Logging,” on page 516.

37.3 Event Codes

A description of the Access Manager event codes is available on the documentation site (http://www.novell.com/documentation/novellaccessmanager/index.html). Scroll to the bottom of page.

37.4 Fixing a Failed Secondary Console

If a secondary administration console gets into a non-reporting state, perhaps caused by hardware failure, its configuration needs to be removed from the primary administration console. As long as it is part of the configuration, other Access Manager devices try to contact it.

If you cannot remove it the normal way, which is by running the uninstall script on the secondary administration console, you can remove it using the troubleshooting page.

1. In the Administration Console, click *Access Manager* > *Auditing* > *Troubleshooting*.
2. In the *Other Known Device Manager Servers* section, select the failed console, then click *Delete*.

All references to the secondary administration console are removed from the configuration database.

37.5 Converting a Secondary Console into a Primary Console

In order for a secondary Administration Console to be converted into a primary Administration Console, a recent backup of the Administration Console must be available. This is necessary in order to restore the certificate authority (CA). This procedure involves changing the master replica and restoring the CA certificates.

WARNING: Perform these steps only if the primary administration console cannot be restored.

If the failed server holds a master replica of any partition, you must use `ndsrepair` to designate a new master replica on a different server in the replica list.

This conversion includes the following tasks:

- Section 37.5.1, "Shutting Down the Server," on page 580
- Section 37.5.2, "Changing the Master Replica," on page 580
- Section 37.5.3, "Restoring CA Certificates," on page 581
- Section 37.5.4, "Deleting Objects from the eDirectory Configuration Store," on page 581
- Section 37.5.5, "Additional Procedures," on page 581

37.5.1 Shutting Down the Server

If your primary Administration Console is running, you must log in as administrator and shut down the server.

1. At the terminal, type `ps aux | grep ndsd`.
2. Take note of the process ID (PID) in the second column.
3. Type `kill -9 <`*`PID`*`>`.

   For example, `kill -9 19124`.
4. Repeat the preceding steps, using `tomcat` instead of `ndsd` in the command.

37.5.2 Changing the Master Replica

1. At the console of one of the servers that shared a replica with the failed server, navigate to the `/opt/novell/eDirectory/bin` directory.

   WARNING: If you use DSRepair with -a or -Ad, some of the advanced options can cause damage to your tree. For more information on these options, refer to the Novell Support Web site, Solution 2938493 (http://support.novell.com/servlet/tidfinder/2938493).

   ---
2. Run `./ndsrepair -P -Ad.`
3. Choose the one available replica.
4. Choose *Designate this server as the new master replica.*

5. Run `ndsrepair -P -Ad` again.
6. Choose the one available replica.
7. Choose *View replica ring*.
8. Select the name of the failed primary server.
9. Choose *Remove this server from replica ring*.

37.5.3 Restoring CA Certificates

From a terminal window on your new primary administration console, run `aminst-certs.sh` from the `/opt/novell/devman/bin` directory.

37.5.4 Deleting Objects from the eDirectory Configuration Store

Several objects representing the failed primary administration console in the configuration store must be manually deleted.

1. Log in to the new Administration Console, then click the *View Objects* icon at the top of the page.

   You can use an LDAP browser instead of the Administration Console.
2. Expand the *novell* container.

   A list of objects is displayed. Delete only the objects that have the name of the failed primary administration console in them. There should be 20 or more objects that need to be deleted.

   The object's name is the hostname of your Linux server during installation. The default name is *linux*.
3. To delete an object, click the object, then click *Delete Object*.
4. In the *novell* container, expand the accessManagerContainer object.
5. Expand *VCDN_Root* > *PartitionsContainer* > *Partition* > *ROMAServerContainer*.
6. Delete the object that has the name of the failed primary administration console.
7. Click the *Roles and Tasks* icon at the top of the page to return to the Access Manager menu.

37.5.5 Additional Procedures

You might need to perform additional steps for the following components:

- "Third Administration Console" on page 582
- "NetWare Access Gateways" on page 582
- "Linux Access Gateways" on page 582
- "SSL VPN" on page 583
- "Identity Servers" on page 584
- "Linux J2EE Agents" on page 584

Third Administration Console

If you installed a third Administration Console used for failover, you must manually perform the following steps on that server:

1. Edit the `vcdn.conf` file in the `/opt/novell/devman/share/conf/` directory.

   In the file you will find a section of XML that looks similar to the following:

   ```
   <vcdnPrimaryAddress>10.1.1.1</vcdnPrimaryAddress>
   ```

   Where 10.1.1.1 represents the failed primary administration console IP address. You must change this IP address to the IP address of the new primary administration console.
2. Restart the Administration Console by entering the following command from the command line interface:

   ```
   /etc/init.d/novell-tomcat4 restart
   ```

NetWare Access Gateways

For each NetWare® Access Gateway imported into the Administration Console, you must perform the following steps:

1. Enter debug mode on the server by entering `debug` and using the password `proxydebug`.
2. Go to the NetWare prompt by pressing Ctrl+Esc if you are using the keyboard. If you are remote via ssh, press Ctrl+Z, then select screen 1.
3. Enter `java -show`, and note the process ID next to `JCCServerImpl`.
4. Enter `java -kill###`, where ### represents the process ID.
5. Edit the `ecc.cfg` file by entering

   ```
   edit sys:\etc\proxy\ecc.cfg
   ```

   Find the section labeled `[jccsettings]` and change the IP address of the line labeled `serveraddress` to the new primary Administration Console.
6. Edit the `settings.properties` file by entering

   ```
   edit sys:\jcc\conf\settings.properties
   ```

   Change the IP address list labeled `remotemgmtip`, removing the IP address of the failed Administration Console. Ensure that the address of the new primary server is listed.
7. Restart the server by entering `appboot` at the NetWare prompt.

Linux Access Gateways

For each Linux Access Gateway imported into the Administration Console, you must perform the following steps:

1. Log in as the `root` user.
2. Open a terminal window and shut down all services by entering the following commands:

   ```
   /etc/init.d/novell-jcc stop
   /etc/init.d/novell-tomcat4 stop
   /etc/init.d/novell-vmc stop
   ```
3. If you are running SSL VPN, enter the following command to stop SSL VPN:

   ```
   /etc/init.d/novell-sslvpn stop
   ```

4 Edit the `config.xml` file by entering

```
vi /var/novell/cfgdb/.current/config.xml
```

  4a Enter `/Remote`, then press Enter.

  In the `IPv4Address` field, change the IP address from the failed administration console to the new primary administration console address.

  4b Enter `/NsureAuditSetting`, then press Enter.

  In the `IPv4Address` field, change the IP address from the failed administration console to the new primary administration console address.

5 Enter `:wq!` to save and exit.

6 Edit the `settings.properties` file by entering

```
vi /opt/novell/devman/jcc/conf/settings.properties.
```

Change the IP address in the `remotemgmtip` list by removing the IP address of the failed administration console. Ensure that the address of the new primary server is listed.

7 Enter `:wq!` to save and exit.

8 Start all services by entering the following commands:

```
/etc/init.d/novell-jcc start
/etc/init.d/novell-tomcat4 start
/etc/init.d/novell-vmc start
/etc/init.d/novell-sslvpn start
```

SSL VPN

For each SSL VPN component imported into the Administration Console, you must perform the following steps:

1 Log in as the `root` user.

2 Open a terminal window and shut down all services by entering the following commands:

```
/etc/init.d/novell-jcc stop
/etc/init.d/novell-tomcat4 stop
/etc/init.d/novell-sslvpn stop
```

3 Edit the `config.xml` file by entering

```
vi /etc/opt/novell/sslvpn/config.xml.
```

4 Enter `/DeviceManagerAddress`, then press Enter.

5 Change the IP address to that of the new primary Administration Console.

6 Enter `:wq!` to save and exit.

7 Edit the `settings.properties` file by entering:

```
vi /opt/novell/devman/jcc/conf/settings.properties
```

Change the IP address list labeled `remotemgmtip` by removing the IP address of the failed administration console. Ensure that the address of the new primary administration console is listed.

8 Enter `:wq!` to save and exit.

9 Start all services by typing the following commands:

```
/etc/init.d/novell-jcc start
/etc/init.d/novell-tomcat4 start
/etc/init.d/novell-sslvpn start
```

If the SSLVPN is no longer functioning, restart the Linux server by entering `reboot`.

Identity Servers

For each Identity Server imported into the Administration Console, you must perform the following steps:

1. Log in as the `root` user.
2. Open a terminal window and shut down all services by entering the following commands:
   ```
   /etc/init.d/novell-jcc stop
   /etc/init.d/novell-tomcat4 stop
   ```
3. Edit the `settings.properties` file by entering
   ```
   vi /opt/novell/devman/jcc/conf/settings.properties
   ```
   Change the IP address list labeled `remotemgmtip` by removing the IP address of the failed administration console. Ensure that the address of the new primary administration console is listed.
4. Enter `:wq!` to save and exit.
5. Start the services by entering the following commands:
   ```
   /etc/init.d/novell-jcc start
   /etc/init.d/novell-tomcat4 start
   ```

Linux J2EE Agents

For each J2EE agent imported into the Administration Console, you must perform the following steps:

1. Log in as the `root` user.
2. Open a terminal window and shut down all services by entering
   ```
   /etc/init.d/novell-jcc stop
   ```
3. Edit the `settings.properties` file by entering:
   ```
   vi /opt/novell/devman/jcc/conf/settings.properties
   ```
   Change the IP address list labeled `remotemgmtip` by removing the IP address of the failed administration console. Ensure that the address of the new primary administration console is listed.
4. Enter `:wq!` to save and exit.
5. Start the services by typing `/etc/init.d/novell-jcc start`.

37.6 Session Conflicts

Do not use two instances of the same browser to simultaneously access the same Administration Console. Browser sessions share settings, which can result in problems when you apply changes to configuration settings. However, you can use two different brands of browsers simultaneously, such as Internet Explorer and Firefox, which makes it possible to avoid the session conflicts.

37.7 Unable to Log In to the Administration Console

If you experience problems logging in to the Administration Console, you might need to restart Tomcat. In a terminal window on the console machine, enter the following command:

```
/etc/init.d/novell-tomcat4 restart
```

If this does not solve the problem, check the `/var/opt/novell/tomcat4/logs/catalina.out` file. Check for the following error:

```
Error Starting up core services.
Application manager is Shutting down the Device Manager suite.
Shutting down Device Manager suite.
```

If you see this error, check the status of eDirectory with the following command:

```
/etc/init.d/ndsd status
```

If the status command returns nothing, you need to manually start eDirectory with the following command:

```
/etc/init.d/ndsd start
```

Then restart tomcat.

```
/etc/init.d/novell-tomcat4 restart
```

37.8 Exception Processing IdentityService_ServerPage.JSP

If you see the message `Exception processing IdentityService_ServerPage.jsp`, it is an indication that the system has run out of available file handles. You need to use the command line to increase the ulimit value (`ulimit -n [new limit]`), which sets the number of open file descriptors allowed.

To set this value permanently, you can create the `/etc/profile.local` file with the ulimit value, such as:

```
ulimit -n 4096
```

You can make changes to `/etc/security/limits.conf` file with a line just to change the limit for a specific user, in this case the `novlwwuser`. You would do this by adding the following line:

```
novlwww soft nofile [new limit]
```

37.9 Backup/Restore Failure Because of Special Characters in Passwords

Administration passwords with special characters such as dollar signs might cause the backup script `/opt/novell/devman/bin/ambkup.sh` to fail. The `ambkup.sh` script creates a command line for the ICE utility, and the special characters might be interpreted by it. If you must use special characters, and this issue arises, modify the `/opt/novell/devman/bin/defbkparm.sh` script so that the special characters are escaped. For example, if the administrator's password is mi$$le, then the field `DS_ADMIN_PWD` should be `mi\$\$le`.

Troubleshooting for the Identity Server and Authentication

38

This section discusses the following topics:

- Section 38.1, "Useful Networking Tools," on page 587
- Section 38.2, "Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors," on page 587
- Section 38.3, "Authentication Issues," on page 594
- Section 38.4, "Translating the Identity Server Configuration Port," on page 596
- Section 38.5, "Problems Reading Keystores after Identity Server Re-installation," on page 598

Identity Server logging information can be found in Section 32.2, "Configuring Identity Server Logging," on page 516 and in Appendix E, "Logging: Using the Custom Content Filter," on page 713.

38.1 Useful Networking Tools

You can use the following tools (Linux and open source) to troubleshoot network problems:

- **netstat:** Displays information related to open ports on your server. Lets you view listeners and various IP addresses, such as the TCP output state.
- **iptables:** See Section 38.4, "Translating the Identity Server Configuration Port," on page 596.
- **netcat:** A networking utility that reads and writes data across network connections, using the TCP/IP protocol. Netcat is useful for checking connectivity with the user store.
- **ldapsearch:** An LDAP search tool useful for the Administration Console and Identity Server. For example, you can generate an LDAP search/bind matching what the Identity Server sends, to confirm whether an issue is with the Identity Server JAR files.
- **tcpdump:** A command line tool for monitoring network traffic. Captures and displays packet headers and matches them against a set of criteria.
- **LDAP Browser/Editor:** Lets you export configuration information to a file, and to confirm that Access Manager objects and attribute values are valid in an AccessManagerContainer. A number of open source versions are available from the Internet.

38.2 Troubleshooting 100101043 and 100101044 Liberty Metadata Load Errors

The Identity Server is the identity provider for the other Access Manager components. The Access Gateways and J2EE Agents have embedded service providers. (The SSL VPN server uses the embedded service provider of the Access Gateway for authentication.) When an Access Gateway or an agent is imported into the Administration Console and a Identity Server configuration is selected for them, a trusted relationship is established with the Identity Server by using test certificates. When you change these certificates or change from using HTTP to HTTPS, you need to make sure that the trusted relationship is re-established. Metadata is used for establishing trusted relationships.

The metadata exchanged between service providers and identity providers contains public key certificates, key descriptors for message signing, a URL for the SSO service, a URL for the SLO (single logout) service, and so on. With Access Manager, this metadata is accessible on both the Identity Server and the Access Gateway. Errors are generated when either the identity provider could not load the service provider's metadata (100101043), or the service provider could not load the metadata of the identity provider (100101044).

If users are receiving either of these errors when they attempt to log in, verify the following:

- Section 38.2.1, "DNS Name Resolution," on page 588
- Section 38.2.2, "Certificate Names," on page 589
- Section 38.2.3, "Certificates in the Required Trust Stores," on page 590
- Section 38.2.4, "Certificates in the Correct Certificate Store," on page 591

If these steps do not solve your problem, try the following:

- Section 38.2.5, "Enable Debug Logging," on page 592
- Section 38.2.6, "Test Whether the Provider Can Access the Metadata," on page 594
- Section 38.2.7, "Manually Create Any Auto-Generated Certificates," on page 594
- For information about metadata validation process and the flow of events that occur when accessing a protected resource on the Access Gateway, see Troubleshooting 100101043 and 100101044 Errors in Access Manager (http://www.novell.com/coolsolutions/appnote/19456.html).

38.2.1 DNS Name Resolution

When the service provider tries to access the metadata on the identity provider, it sends the request to the hostname defined in the base URL configuration of the Identity Server. The base URL in the Identity Server configuration is used to build all the metadata end points.

To view the metadata of the Identity Server with a DNS name of idpcluster.lab.novell.com, enter the following URL:

```
https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

Scan through the document and notice the multiple references to https://idpcluster.lab.novell.com/... You should see lines similar to the following:

```
<md:SoapEndpoint>
   https://idpcluster.lab.novell.com:8443/nidp/idff/soap
</md:SoapEndpoint>

<md:SingleLogoutServiceURL>
   https://idpcluster.lab.novell.com:8443/nidp/idff/slo
</md:SingleLogoutServiceURL>

<md:SingleLogoutServiceReturnURL>
   https://idpcluster.lab.novell.com:8443/nidp/idff/slo_return
</md:SingleLogoutServiceReturnURL>
```

The embedded service provider of the Access Gateway must be able to resolve the idpcluster.lab.novell.com hostname of the Identity Server. To test that it is resolvable, send a ping command with the hostname of the Access Gateway. For example, from the Identity Server:

```
ping idpcluster.lab.novell.com
```

The same is true for the Identity Server. It must be able to resolve the hostname of the Access Gateway. To discover the URL for the Access Gateway metadata:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *Reverse Proxy/Authentication*.
2. View the *Embedded Service Provider* section.

   The URL of the metadata is displayed in this section.

To view the metadata, enter the displayed URL. Scan through the document and notice the multiple references to the hostname of the Access Gateway. You should see lines similar to the following. In these lines, the hostname is ag1.provo.novell.com.

```
<md:SoapEndpoint>
   http://ag1.provo.novell.com:80/nesp/idff/spsoap
</md:SoapEndpoint>

<md:SingleLogoutServiceURL>
   http://ag1.provo.novell.com:80/nesp/idff/spslo
</md:SingleLogoutServiceURL>

<md:SingleLogoutServiceReturnURL>
   http://ag1.provo.novell.com:80/nesp/idff/spslo_return
</md:SingleLogoutServiceReturnURL>
```

To test that the Access Gateway can resolve the hostname of the Identity Server, send a ping command with the hostname of the Identity Server. For example, from the Access Gateway:

```
ping ag1.provo.novell.com
```

To view sample log entries that are logged to the catalina.out file when a DNS name cannot be resolved, see "The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server" on page 592.

38.2.2 Certificate Names

Make sure the certificates for the Identity Server and the embedded service provider match the hostnames defined in the metadata URL (see Section 38.2.1, "DNS Name Resolution," on page 588).

When the Identity Server and the Access Gateway are enabled for HTTPS, all communication to these devices require that the devices send back a server certificate. Not only must the certificate be assigned to the appropriate device, but the subject name of the device certificate must match the hostname of the device it is assigned to.

To verify the certificate name of the Identity Server certificate:

1. In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit*.
2. Click the *SSL Certificate* icon.

   The NIDP-connector keystore is displayed
3. Verify that the subject name of the certificate matches the DNS name of the Identity Server.
   - If the names match, a certificate name mismatch is not causing your problem.

  - If the names do not match, you need to either create a certificate that matches or import one that matches. For information on how to create a certificate for the Identity Server, see "Configuring Secure Communication on the Identity Server" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

To verify the certificate name of the Access Gateway certificate:

1. In the Administration Console, click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.
2. Read the alias name of the server certificate, then click the *Server Certificate* icon.
3. Verify that the Subject name of the server certificate matches the published DNS name of the proxy service of the Access Gateway.
   - If the names match, a certificate name mismatch is not causing your problem.
   - If the names do not match, you need to either create a certificate that matches or import one that matches. For information on how to create an Access Gateways certificate, see Chapter 14, "Configuring the Access Gateway for SSL," on page 239.

To view sample log entries that are logged to the `catalina.out` file when the certificate has an invalid name, see "The Server Certificate Has an Invalid Subject Name" on page 593.

38.2.3 Certificates in the Required Trust Stores

Make sure that the issuers of the Identity Server and embedded service provider certificates are added to the appropriate trusted root containers.

When the server certificates are sent from the identity provider to the service provider client, and from the service provider to the identity provider client, the client needs to be able to validate the certificates. Part of the validation process is to confirm that the server certificate has been signed by a trusted source. To do this, the issuers of the server certificate (intermediate and trusted roots) must be imported into the correct trusted root stores:

- The intermediates and trusted roots of the embedded service provider certificate must be imported into the NIDP-Truststore.
- The intermediates and trusted roots of the Identity Server certificate must be imported into the ESP Trust Store.

If you use certificates generated by the Administration Console CA, the trusted root certificate is the same for the Identity Server and the embedded service provider. If you are using external certificates, the trusted root certificate might not be the same, and there might be intermediate certificates that need to be imported.

To verify the trusted root certificates:

1. In the Administration Console, click *Access Manager* > *Certificates*.
2. Determine the issuer of the Identity Server certificate and the embedded service provider certificate:
   - **2a** Click the name of the Identity Server certificate, note the name of the Issuer, then click *Close*.
   - **2b** Click the name of the embedded service provider certificate, note the name of the Issuer, then click *Close*.
3. To verify the trusted root for the Identity Server, click *Trusted Roots* > *NIDP-truststore*.

4. Scan for a certificate subject that matches the issuer of the embedded service provider certificate, then click its name.
    - If the Issuer has the same name as the Subject name, then this certificate is the root certificate.
    - If the Issuer has a different name than the Subject name, the certificate is an intermediate certificate in the chain. Click *Close*, and make sure another certificate in the trust store is the root certificate. If it isn't there, you need to import it and any other intermediate certificates between the one you have and the root certificate.
5. To verify the trusted root for the embedded service provider, click *Trusted Roots* > *ESP Trust Store*.
6. Scan for a certificate subject that matches the issuer of the Identity Server certificate, then click its name.
    - If the Issuer has the same name as the Subject name, then this certificate is the root certificate.
    - If the Issuer has a different name than the Subject name, the certificate is an intermediate certificate in the chain. Click *Close*, and make sure another certificate in the trust store is the root certificate. If it isn't there, you need to import it and any other intermediate certificates between the one you have and the root certificate.
7. (Optional) If you have clustered your Identity Servers and Access Gateways and you are concerned that not all members of the cluster are using the correct trusted root certificates, you can re-push the certificates to the cluster members.
    - **7a** Click *Access Manager* > *Auditing* > *Troubleshooting* > *Certificates*.
    - **7b** Select the Trust Store of your Identity Servers and Access Gateways, then click *Re-push certificates*.
    - **7c** Update the Identity Severs and Access Gateways.
    - **7d** Check the command status of each device to ensure that the certificate was pushed to the device. From the Identity Servers page or the Access Gateways page, click the *Commands* link.

To view sample log entries that are logged to the `catalina.out` file when a trusted root certificate is missing, see "Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers" on page 593.

38.2.4 Certificates in the Correct Certificate Store

Make sure that the server certificates are added to the correct certificate store. In other words, the Identity Server certificate must be added to the NIDP-connector store, and the embedded service provider certificate must be added to the Proxy Key Store.

1. In the Administration Console, click *Access Manager* > *Certificates*.
2. Click *NIDP-connector*.
3. Verify that the certificate is the correct certificate for the Identity Server. The subject name should match the hostname of the Identity Server. If it doesn't match, replace it.
4. Click *Close*, then *Proxy Key Store*.

5 Verify that the certificate is the correct certificate for the embedded service provider. The subject name should match the published DNS name of the proxy service on the Access Gateway. If it doesn't match, add one that does match.

6 Click *Close*.

38.2.5 Enable Debug Logging

You can enable Identity Server logging to dump more verbose Liberty information to the `catalina.out` file on both the Identity Server and the embedded service provider of the Access Gateway.

1 In the Administration Console, click *Access Manager* > *Identity Servers* > *Edit* > *Logging*.

2 Select *Enabled* for *File Logging* and *Echo to Console*.

3 In the *Component File Logger Levels* section, set *Application* and *Liberty* to a *debug* level.

4 Click *OK*, update the Identity Server, then update the Access Gateway

After enabling and applying the changes, duplicate the issue once more to add specific details to the `catalina.out` file for the issue. If the error is the 100101044 error, look at the `catalina.out` file on the embedded service provider for the error code; if the error is the 100101043 error, look at the `catalina.out` file on the Identity Server for the error code.

Below are a few typical entries illustrating the most common problems. They are from the `catalina.out` file of the embedded service provider:

- "The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server" on page 592
- "Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers" on page 593
- "The Server Certificate Has an Invalid Subject Name" on page 593

The Embedded Service Provider Cannot Resolve the Base URL of the Identity Server

When the embedded service provider cannot resolve the DNS name of the Identity Server, the metadata cannot be loaded and a hostname error is logged. In the following entries, the embedded service provider cannot resolve the idpcluster.lab.novell.com name of the Identity Server.

```
<amLogEntry> 2007-08-06T16:24:56Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#2CA1168DF7343A42C7879
E707C51A03C: ESP is requesting metadata from IDP https://
idpcluster.lab.novell.com/nidp/idff/metadata </amLogEntry>

<amLogEntry> 2007-08-06T16:24:56Z SEVERE NIDS IDFF: AM#100106001:
AMDEVICEID#esp-09C720981EEE4EB4: Unable to load metadata for Embedded
Service Provider: https://idpcluster.lab.novell.com/nidp/idff/
metadata, error: AM#300101046: AMDEVICEID#esp-09C720981EEE4EB4::
Attempted to connect to a url with an unresolvable host name
</amLogEntry>

<amLogEntry> 2007-08-06T16:24:56Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#2CA1168DF7343A42C7879
E707C51A03C: Error on session id 2CA1168DF7343A42C7879E707C51A03C,
error 100101044-esp-09C720981EEE4EB4, Unable to authenticate.
```

```
AM#100101044: AMDEVICEID#esp-09C720981EEE4EB4:: Embedded Provider
failed to load Identity Provider metadata </amLogEntry>
```

Trusted Roots Are Not Imported into the Appropriate Trusted Root Containers

When the trusted roots are not imported into the appropriate trusted root containers, a certificate exception is thrown and an untrusted certificate message is logged. In the following log entries, the embedded service provider is requesting metadata from the Identity Server, but the embedded service provider does not trust the Identity Server certificate because the trusted root of the issuer of the Identity Server certificate is not in the embedded service provider's trusted root container.

```
<amLogEntry> 2007-08-05T16:07:53Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D13
9D33E5324F98F: ESP is requesting metadata from IDP https://
idpcluster.lab.novell.com/nidp/idff/metadata </amLogEntry>

<amLogEntry> 2007-08-05T16:07:53Z SEVERE NIDS IDFF: AM#100106001:
AMDEVICEID#esp-09C720981EEE4EB4: Unable to load metadata for Embedded
ServiceProvider: https://idpcluster.lab.novell.com/nidp/idff/metadata,
error: java.security.cert.CertificateException: Untrusted Certificate-
chain </amLogEntry>

<amLogEntry> 2007-08-05T16:07:53Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983 B08C28D35221D139
D33E5324F98F: Error on session id D983B08C28D35221D139D33E5324F98F,
error 100101044-esp-09C720981EEE4EB4, Unable to authenticate.
AM#100101044: AMDEVICEID#esp-09C720981EEE4EB4:: Embedded Provider
failed to load Identity Provider metadata </amLogEntry>
```

The Server Certificate Has an Invalid Subject Name

When the certificate has an invalid subject name, the handshake fails. In the log entries below, the embedded service provider is requesting metadata from the Identity Server. The server certificate name does not match, so the embedded service provider is unable to authenticate and get the metadata necessary to establish the trusted relationship.

```
<amLogEntry> 2007-07-05T16:07:53Z INFO NIDS Application: AM#500105024:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D139D33
E5324F98F: ESP is requesting metadata from IDP
https://idpcluster.lab.novell.com/nidp/idff/metadata </amLogEntry>

<amLogEntry> 2007-07-05T16:07:53Z SEVERE NIDS IDFF: AM#100106001:
AMDEVICEID#esp-09C720981EEE4EB4: Unable to load metadata for Embedded
Service Provider: https://idpcluster.lab.novell.com/nidp/idff/
metadata, error: Received fatal alert: handshake_failure </amLogEntry>

<amLogEntry> 2007-07-05T16:07:53Z INFO NIDS Application: AM#500105039:
AMDEVICEID#esp-09C720981EEE4EB4: AMAUTHID#D983B08C28D35221D139D33
E5324F98F: Error on session id D983B08C28D35221D139D33E5324F98F, error
100101044-esp-09C720981EEE 4EB4, Unable to authenticate. AM#100101044:
AMDEVICEID#esp-09C720981EEE4EB4: : Embedded Provider failed to load
Identity Provider
metadata </amLogEntry>
```

38.2.6 Test Whether the Provider Can Access the Metadata

To test whether the metadata is available for download, enter the metadata URL of the identity provider and service provider. If the DNS name of the identity provider is idpcluster.lab.novell.com, open a browser and enter the following URL:

```
https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

Because the Linux Access Gateway does not have a graphical interface, you need to use the `curl` command to test whether the Access Gateway can access the metadata of the Identity Server. If the NDS name of the identity provider is idpcluster.lab.novell.com, enter the following command from the Access Gateway machine:

```
curl -k https://idpcluster.lab.novell.com:8443/nidp/idff/metadata
```

To test whether the Identity Server can access the metadata URL of the Access Gateway, open a browser on the Identity Server machine. If the published DNS name of service provider is www.aleris.net, enter the following URL:

```
https://www.aleris.net/nesp/idff/metadata
```

38.2.7 Manually Create Any Auto-Generated Certificates

Occasionally, there are issues where the subject name was auto-generated and the entire configuration appears to be correct, but the 100101044/100101043 error is still reported. Delete the auto-generated certificate and manually re-create the server certificate, making sure that it is added to the relevant devices and stores.

38.3 Authentication Issues

This section discusses the following issues that occur during authentication:

- Section 38.3.1, "General Authentication Troubleshooting Tips," on page 594
- Section 38.3.2, "Slow Authentication," on page 595
- Section 38.3.3, "Basic Authentication Fails with an eDirectory User Store," on page 595
- Section 38.3.4, "Federation Errors," on page 595
- Section 38.3.5, "Mutual Authentication Troubleshooting Tips," on page 595
- Section 38.3.6, "Browser Hangs in an Authentication Redirect," on page 596

38.3.1 General Authentication Troubleshooting Tips

- Use LAN traces to check requests, responses, and interpacket delay times.
- In the user store logs, confirm that the request arrived. Check for internal errors.
- Check the user store health and replica layout. See TID 3066352.
- Ensure that the user exists in the user store and that the context is defined.
- Check the properties of the class and method. For example, the search format on the properties must match what you've defined on a custom login page. You might be asking for a name/password login, but the method specifies e-mail login criteria.
- Enable authentication logging options. (*Identity Servers* > *Edit* > *Logging* > *Novell Audit Logging*.)

- Ensure that the authentication contract matches the base URL scheme. For example, is SSL used across all components?

38.3.2 Slow Authentication

The following configuration problems can cause slow authentication:

- If authentication is taking up to a minute per user, verify that your DNS server has been enabled for reverse lookups. The JNDI module in the Identity Server sends out a request to resolve the IP address of the LDAP server to a DNS name. If your DNS server is not enabled for reverse lookups, it takes 10 seconds for this request to fail before the Identity Server can continue with the authentication request.
- If your user store resides on SUSE® Linux Enterprise Server 10, which installs with a firewall, you must open TCP 524. For more information about the ports that must be open when a firewall separates the user store from other Access Manager components, see "Setting Up Firewalls" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

38.3.3 Basic Authentication Fails with an eDirectory User Store

You are not required to specify a search context with eDirectory™. However, when a search context is not specified, the entire directory tree is searched for the specified username. If the username is present in more than one context, authentication fails.

When using eDirectory as the user store, you should ensure that all usernames in the directory are unique, and you should also specify a search context. Otherwise, every authentication request generates a request to search the entire directory. For a small directory, this might not be significant, but for a large directory, it could take a significant amount of time.

38.3.4 Federation Errors

- Most errors that occur during federation occur because of time synchronization problems between servers. Ensure that all of your servers involved with federation have their time synchronized within one minute.
- When the user denies consent to federate after clicking a Liberty link and logging in at the identity provider, the system displays an error page. The user should acknowledge that federation consent was denied and return to the service provider login page. This is the expected behavior when a user denies consent.

38.3.5 Mutual Authentication Troubleshooting Tips

- LAN traces:
  - Check the SSL handshake and look at trusted root list that was returned.
  - The client certificate issuer must be in the identity provider certificate store and be applied to all the devices in a cluster.
  - Ensure that the user exists and meets the authentication criteria. As the user store administrator, you can search for a subject name (or certificate mapping attributes defined) to locate a matching user.

- Enable the *Show Certificate Errors* option on the Attributes page for the X.509 authentication class. (*Identity Servers > Servers > Edit > Local > Classes > [x.509] > Properties*.) Enabling this option provides detailed error messages on the login browser, rather than generic messages.
- Ensure that the certificate subject name matches the user you log in with, if you are chaining methods.
- Use NTRadPing to test installations.
- Verify that the correct UDP port 1812 is specified.
- Verify that the RADIUS server can accept requests from the IDP server. This might require the NAS-IP-Address attribute along with credentials.
- Verify that the user exists in the user store if multiple methods are added to a contract.
- Verify if user authentication works independent of Access Manager.
- Verify that the NMAS™ server is local and no tree walks are occurring across the directory.
- Ensure that the NMAS_LOGIN_SEQUENCE property is defined correctly.

38.3.6 Browser Hangs in an Authentication Redirect

If the browser hangs when the user attempts to authenticate at an identity provider, determine whether a new authentication contract was created and set as the default contract on the Identity Server. If this is the case and you have an Access Gateway resource set to accept any contract from the identity provider, you should navigate to the *Overview* tab for the protected resource and specify *Any* again in the *Contract* drop-down menu. Then click *OK*, then update the Access Gateway.

38.4 Translating the Identity Server Configuration Port

If your Identity Server must communicate with an external Identity Server through a firewall, you must either set up a hole in your firewall for TCP ports 8080 or 8443 (default ports used respectively for non secure and secure communication with Identity Server), or configure the Identity Server service to use TCP port 80 or 443.

The Identity Server service (hosted on Tomcat) runs as a non-privileged user and cannot therefore bind to ports below 1024. In order to allow requests to port 80/443 while Tomcat is listening on 8080/8443, the preferred approach is to use iptables to perform a port translation. Assuming HTTPS on port 443 is used, perform the following procedure. Similar steps apply to using HTTP on port 80 if a non secure channel is required.

1. In the Administration Console, click *Identity Server* > *Servers* > *Edit*, and configure the base URL with HTTPS as protocol, and the TCP Port as 443.
2. At a terminal window, log in as the `root` user.
3. Create a file to hold the iptables rule and place it in the `/etc/init.d` directory.

   For example, `/etc/init.d/Redirect`. An example of a redirect startup file for this purpose might be:

   ```
   #!/bin/sh
   # Copyright (c) 2008 Novell, Inc.
   # All rights reserved.
   #
   #! /bin/sh
   ```

```
#! /etc/init.d/idp_8443_redirect
# ### BEGIN INIT INFO
# Provides: idp_8443_redirect
# Required-Start: SuSEfirewall2_setup $network $local_fs
# Required-Stop:
# Default-Start: 2 3 5
# Default-Stop: 0 1 6
# Description: Redirect 8443 to 443 for Novell IDP
### END INIT INFO #

# Environment-specific variables.
IPT_BIN=/usr/sbin/iptables
INTF=eth0
ADDR=10.10.0.1

. /etc/rc.status

# First reset status of this service
rc_reset

case "$1" in
    start)
        echo -n "Starting IP Port redirection"
        $IPT_BIN -t nat --flush
        $IPT_BIN -t nat -A PREROUTING -i $INTF -p tcp --dport 80
-j DNAT --to ${ADDR}:8080
        $IPT_BIN -t nat -A PREROUTING -i $INTF -p tcp --dport 443
-j DNAT --to ${ADDR}:8443
        rc_status -v
        ;;
    stop)
        echo -n "Flushing all IP Port redirection rules"
        $IPT_BIN -t nat --flush
        rc_status -v
        ;;
    restart)
        $0 stop
        $0 start
        rc_status
        ;;
    *)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac
rc_exit
```

For more information about init scripts in SUSE Linux Enterprise Server, see 20.2.2 Init Scripts (http://www.novell.com/documentation/sles10/index.html?page=/documentation/sles10/sles_admin/data/sec_boot_init.html) in the *SUSE Linux Enterprise Server 10 Installation and Administration Guide* (http://www.novell.com/documentation/sles10/index.html).

4 Modify the environment-specific variables found in the following lines:

```
# Environment-specific variables.
IPT_BIN=/usr/sbin/iptables
```

```
INTF=eth0
ADDR=10.10.0.1
```

5. Ensure that your redirect script has execute rights. Use CHMOD as appropriate. For example:

   ```
   chmod 755 Redirect
   ```

6. Ensure that the iptables rule is active after rebooting:

   6a In YaST, click *System* > *Runlevel Editor* [or *System Services (Runlevel)*].

   6b Select the `Redirect` service, and enable it.

7. (Conditional) On SLES 9.x when you enable the redirect script in the Runlevel editor, it automatically enables 3 startup scripts for the SuSEfirewall2. You need to disable the `SuSEfirewall2_final` script because it overwrites the rules in your `Redirect` script.

   7a In YaST click *System* > *Runlevel Editor*.

   7b Select the `SuSEfirewall2_final` script, and disable it.

8. To verify that the script is running, enter the following command:

   ```
   iptables -t nat --list
   ```

   If it is running, the output should contain lines similar to the following:

   ```
   Chain PREROUTING (policy ACCEPT)
   target prot opt source   destination
   DNAT   tcp  --  anywhere anywhere tcp dpt:http to:10.10.0.1:8080
   DNAT   tcp  --  anywhere anywhere tcp dpt:https to:10.10.0.1:8443
   ```

IMPORTANT: This simple solution only works if you are not using iptables to translate ports of other applications or Access Manager components. For a solution that works with multiple components, see NAM Filters for iptables Commands (http://www.novell.com/communities/node/4029/nam-filters-iptables-commands).

38.5 Problems Reading Keystores after Identity Server Re-installation

This can occur if you replace a hard drive and incorrectly reinstall the Identity Server. See "Reinstalling an Identity Server to a New Hard Drive" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide* for the correct procedure.

Troubleshooting Access Manager Policies

39

This section discusses the following topics:

39.1 Turning on Logging for Policy Evaluation

Policy evaluation for roles occurs at the Identity Server. For Authorization and Identity Injection policies, policy evaluation occurs on the embedded service provider where the policy is enabled.

For Form Fill policies, the evaluation and logging is done by the embedded service provider and the proxy service. To set the logging level on the Access Gateway for the proxy service, see the following:

- "Linux Access Gateway Logs" on page 669
- "Enabling Form Fill Logging" on page 614

Logging for the policy evaluation done by embedded service providers is controlled by the log settings of the Identity Server configuration. To enable this type of logging:

1. Click *Access Manager* > *Identity Servers* > *Edit* > *Logging*.

   If you have set up more than one Identity Server configuration, make sure you select the configuration to which the other Access Manager components have been assigned.
2. Select *Enabled* for *File Logging*.
3. Select to echo the trace messages to the console.
   - For a Linux Access Gateway, this sends the messages to the `catalina.out` file.
   - For a NetWare® Access Gateway, this sends the messages to the NetWare console.
   - For the Linux Identity Server, this sends the messages to the `catalina.out` file.
4. (Optional) Specify a path for the Identity Server log files.

   If you have a mixed platform environment (for example, the Identity Server is installed on Linux and the Access Gateway is on NetWare), do not specify a path.

5. For policy evaluation tracing, set the *Application* level to *info* in the *Component File Logger Levels* section.

   If you are only troubleshooting polices at this time, do not select any other options. This reduces the amount of information recorded in the log files.

   To see the policy SOAP messages, you need to set the *Application* level to *config*.

6. Update the Identity Server.

7. Click *Auditing* > *General Logging*.

   - For role evaluation traces, view the Identity Server `catalina.out` file.

     If your Identity Servers are clustered, you need to look at the file from each Identity Server.

   - For Authorization, Form Fill, and Identity Injection evaluation traces, view the log file of the embedded service provider of the device that is protecting the resource.

     - For a Linux Access Gateway, this is the `catalina.out` file of the Access Gateway where the protected resource is defined. If the Linux Access Gateway is part of a group, you need to look at this file from each Access Gateway in the group.

       The actual ESP log file is not displayed in the list. To view this file, which contains only ESP log messages, see the `nidp.*.xml` files in the `/var/ops/novell/tomcat4/logs` directory (or the directory you specified in Step 4). Depending upon how you have configured *File Wrap*, the * portion of the filename contains the month, the week, the day, and the hour.

     - For a NetWare Access Gateway, the file is not displayed in the list. To view the trace messages, you need to go to the system console or view the `nipd.*.xml` file in the `sys:\tomcat\4\webapps\nesp\WEB-INF\logs` directory. Depending upon how you have configured *File Wrap*, the * portion of the filename contains the month, the week, the day, and the hour.

       To view the `nipd.*.xml` file, you need to enable FTP or SSH and copy the file.

     - For a J2EE Agent, see "Viewing Log Files" in the *Novell Access Manager 3.0 SP3 J2EE Agent Guide*.

8. To understand what you are looking for in the log file, continue with one of the following:

   - Section 39.2, "Understanding Policy Evaluation Traces," on page 600 if you set *Application* level to *info*.
   - Section 39.9, "Policy Evaluation: Access Gateway Devices," on page 625 if you set *Application* level to *config*.

39.2 Understanding Policy Evaluation Traces

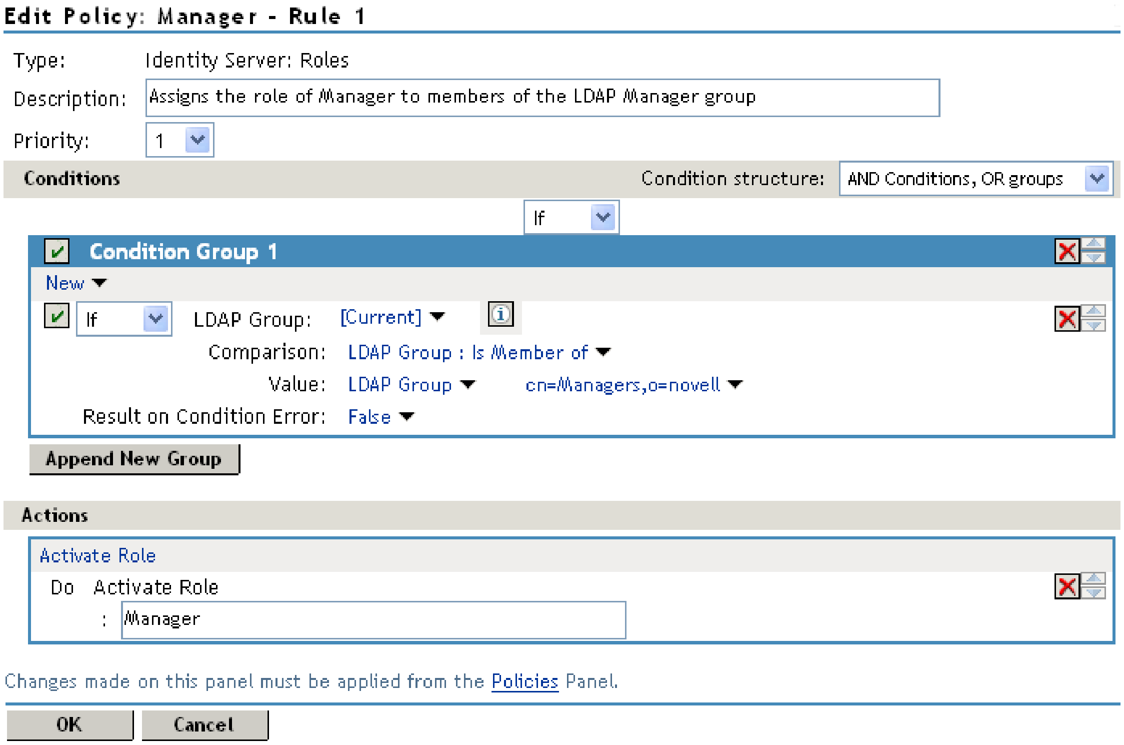
39.2.1 Format

A policy log entry starts with the standard log entry elements: `<amLogEntry>` followed by the correlation tags. (For information about correlation tags, see Section 42.2.1, “Understanding the Correlation Tags in the Log Files,” on page 674.) The following log entry is a trace of an evaluation of a Role policy:

```
<amLogEntry> 2007-06-07T21:40:25Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29:
AMAUTHID#503EFFA4BC21ACA307796EC7D96E5532: IDP RolesPep.evaluate(),
policy trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
   ~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
   ~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VC
DN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc
),Policy=(Manager),Rule=(1::RuleID_1181251958207),Action=(AddRole::Act
ionID_1181252224665)~AdditionalRole(6601):unknown():Manager:~~~Success
(0)
 </amLogEntry>
```

The Role policy evaluated in this entry has the following definition:

Figure 39-1 *Manager Policy Definition*



Edit Policy: Manager - Rule 1

Type: Identity Server: Roles

Description: Assigns the role of Manager to members of the LDAP Manager group

Priority: 1

Conditions Condition structure: AND Conditions, OR groups

If

☑ **Condition Group 1**

New

☑ If LDAP Group: [Current]

Comparison: LDAP Group : Is Member of

Value: LDAP Group cn=Managers,o=novell

Result on Condition Error: False

Actions

Activate Role

Do Activate Role

: Manager

Changes made on this panel must be applied from the Policies Panel.

The following sections use this policy and its trace to explain the information contained within each line of a policy trace. The policy trace part of the entry starts with a `policy trace:`, which is followed by one of the following types:

- RL - Rule List Evaluation Result
- RU - Rule Evaluation Result
- CS - Condition Set Evaluation Result
- CO - Condition Evaluation Result
- PA - Policy Action Initiation
- PC - Policy Action Completion

Elements within a type are separated from each other with the tilde (~) character. If an element does not have a value, no value is inserted, which results in two or more tildes between values. Two tildes means one element didn't have a value, three tildes means that two elements didn't have values, and so forth.

Rule List Evaluation Result

A RL trace has the following fields:

```
~<RuleListID>~~~~~<RuleCount>~~~<Result>
```

A RL trace looks similar to the following:

```
~~RL~1~~~~Rule Count: 1~~Success(67)
```

Table 39-1 describes the fields found in a RL trace.

***Table 39-1** Fields in a Rule List Trace*

Element	Description
`<RuleListID>`	The identifier assigned to the rule list. In the sample RL trace, this is 1.
`<RuleCount>`	The number of rules defined for the policy. In the sample RL trace, this is `Rule Count: 1`, indicating that there is one rule in the policy.
`<Result>`	A string followed by a number that specifies the result of the evaluation. See "Policy Result Values" on page 607. In the sample RL trace, this is `Success(67)`, indicating success.

Rule Evaluation Result

A RU trace has the following fields:

```
~<RuleID>~<ParentPolicyName>~<ConditionSetJoinType>~~<ConditionSetCoun
t:ActionCount>~~<Result>
```

A RU trace looks similar to the following:

```
~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
```

Table 39-2 describes the fields of a Rule Evaluation Result trace.

***Table 39-2** Fields in a Rule Evaluation Result Trace*

Element	Description
`<RuleID>`	The identifier assigned to the rule. In this sample RU trace, this element is set to `RuleID_1181251958207`.
`<ParentPolicyName>`	The name of the parent policy to which the rule is assigned. In this sample RU trace, this element is set to `Manager`.
`<ConditionSetJoinType>`	The type of joining that occurs between conditions and condition sets. It is set to one of the following: ◆ **CNF:** Indicates that sets are ANDed and conditions within a condition group are ORed. ◆ **DNF:** Indicates that sets are ORed and conditions within a condition group are ANDed. In the sample RU trace, this element is set to `DNF`.
`<ConditionSetCount:ActionCount>`	The number of condition sets and actions defined for this rule. In the sample RU trace, this is 1:1, for one condition set and one action.
`<Result>`	A string followed by a number that specifies the result of the evaluation. See "Policy Result Values" on page 607. In the sample RU trace, this is `Success(67)`, indicating that the rule was successfully evaluated.

Condition Set Evaluation Result

A CS trace has the following fields

```
~<ConditionSetID>~<JoinType>~<NOT>~<ConditionCount>~~<Result>
```

A CS trace looks similar to the following:

```
~~CS~1~~ANDs~~1~~True(69)
```

Table 39-3 describes the fields in a Condition Set trace.

***Table 39-3** Fields in a Condition Set Trace*

Element	Description
`<ConditionSetID>`	The identifier assigned to the condition set. Rules can have multiple condition sets. In this sample CS trace, this is 1, for the first and only condition set defined for the rule.

Element	Description
`<JoinType>`	Specifies how the condition results are combined, if there are multiple condition sets. Possible values include `ANDs` and `ORs`. In this sample CS trace, this is `ANDs`.
`<NOT>`	The string `NOT` if the result was negated prior to reporting; otherwise the field has no value. This is the *If Not* option when creating a condition group. In the sample CS trace, the condition group was not negated, therefore the field is not present.
`<ConditionCount>`	The number of conditions defined in the condition group. In the sample CS trace, this element has the value of 1.
`<Result>`	A string followed by a number that specifies the result of the evaluation. See "Policy Result Values" on page 607. In the sample CS trace, this is True (69), indicating that the condition evaluated to True.

Condition Evaluation Result

A CO trace has the following fields:

```
~<ConditionID>~<LHSOperand>~<Operator>~<RHSOperand>~<NOT>~<Result>[~<R
esultOnError>]
```

A CO trace looks similar to the following:

```
~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
```

Table 39-4 describes the fields in a Condition trace.

Table 39-4 *Fields in a Condition Trace*

Element	Description
`<ConditionID>`	The identifier assigned to the conditions in the condition group. The first condition is assigned 1. In the sample CO trace, this is 1.

Element	Description
`<LHSOperand>`	The enumerative value and parameter list of the left operand. It is the first value specified for the comparison and has the following format: `<Condition Name(Data ID)>: <Parameter> : <Value>` The Condition Name is the string assigned to the condition type specified in the policy. The Data ID is a numerical value assigned to the condition type. `<Parameter>` contains one of the following strings: ◆ `no-param` when no parameters are specified for the operand, followed by a colon, followed by one of the following: the value, `no-value`, or `hidden-value` when the value contains sensitive information. ◆ `hidden-param` followed by a colon, and then `hidden-value`. This string is used when both the parameter and its value contain sensitive information. In the sample CO trace, this is `LdapGroup(6645):no-param:hidden-value`. LdapGroup is the string for the LDAP Group condition. The policy specified *[Current]*, so no parameters were specified. The groups that the user belongs to are considered sensitive data, so the log file displays `hidden-value` for the names of the groups.
`<Operator>`	The display name of the comparison operator. In the sample CO trace, this is `ldap-group-is-member-of`. In the policy, this is displayed as *LDAP Group: Is Member of*.
`<RHSOperand>`	The enumerative value and parameter list of the right operand. It is the second value specified for the comparison and has the same format as the `<LHSOperand>`. In the sample CO trace, this is `SelectedLdapGroup(66455):hidden-param:hidden-value`. The actual policy specifies LDAP Group as the parameter, and the value is the DN of the group.
`<NOT>`	The string `NOT` if the result was negated prior to reporting; otherwise the field has no value. This is the *If Not* option when creating a condition. In the sample CO trace, this condition result was not negated, therefore the field is represented by a tilde.
`<Result>`	A string followed by a number that specifies the result of the comparison. See "Policy Result Values" on page 607. In the sample CO trace, this is True (69), indicating that the condition evaluated to True—the user is a member of the specified LDAP group.
`<ResultOnError>`	A string describing the error that occurred. This is an optional field that only appears when the condition evaluation results in an error. The sample CO trace did not result in an error, so it has no string.

Policy Action Initiation

A PA trace has the following fields

```
~<ActionID>~<TraceString1>~<TraceString2>~<TraceString3>~<Result>
```

A PA trace looks similar to the following:

```
~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
```

Table 39-5 describes the fields in a Policy Action trace.

Table 39-5 *Fields in a Policy Action Trace*

Element	Description
`<ActionID>`	The identifier assigned to the action. In the sample PA trace, this is `ActionID_1181252224665`.
`<TraceString1>`	The message specified with the action. In the sample PA trace, this is `AddRole`.
`<TraceString2>`	The second part of the specified message. In the sample PA trace, this is `Manager`.
`<TraceString3>`	The third part of the specified message. In the sample PA trace, this field has no value and is not present.
`<Result>`	A string followed by a number that specifies the result of the assigning the action. See "Policy Result Values" on page 607. In the sample PA trace, this is `Success(0)` and indicates that the action was successfully assigned to the user.

Policy Action Completion

A PC trace has the following fields

```
~<ActionID>~<ActionName>~<ActionParmeters>~~~<Result>[~<ActionError>]
```

A PC trace looks similar to the following:

```
~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VC
DN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc
),Policy=(Manager),Rule=(1::RuleID_1181251958207),Action=(AddRole::Act
ionID_1181252224665)~AdditionalRole(6601):unknown():Manager:~~~Success
(0)
```

Table 39-6 describes the fields in a Policy Action Completion trace.

Table 39-6 *Fields in a Policy Action Completion Trace*

Element	Description
`<ActionID>`	The ID assigned to the action. In the sample PC trace, this is `ActionID_1181252224665`.

Element	Description
`<ActionName>`	The fully distinguished name of the action. In the sample PC trace, the action has the following parts in its name: ◆ Document=(ou=xpemlPEP,ou=mastercdn, ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer, ou=VCDN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc) ◆ Policy=(Manager) ◆ Rule=(1::RuleID_1181251958207) ◆ Action=(AddRole::ActionID_1181252224665)
`<ActionParmeters>`	A list of the action parameters passed to the action handler. In this sample PC trace, the Role policy has an action and a parameter. The value of this element is `AdditionalRole(6601):unknown(): Manager:`
`<Result>`	A string followed by a number that specifies the result. See "Policy Result Values" on page 607. In the sample PC trace, this is `Success(0)` and indicates success.
`<ActionError>`	A string describing the error that occurred when invoking the action. This is an optional field that only appears when the Result field contains an error code. The sample PC trace did not result in an error, so it has no string.

39.2.2 Policy Result Values

The last field in a trace string is the `<result>` field. Table 39-7 lists the possible values:

***Table 39-7** Result Values from Policy Traces*

Value	Name	Description
0	Success	The policy evaluation was successful.
1	Error: No memory	The system is out of memory.
2	Error: Bad data	The data sent for evaluation is invalid.
3	Error: Configuration initialization	An error was detected during the policy configuration processing.
4	Error: General failure	An error was detected during policy processing.
5	Pending	The policy processing is in progress.
64	Permit	The rule produced a Permit action.
65	Deny	The rule produced a Deny action.

Value	Name	Description
66	Obligation	The rule triggered an obligation, indicating that additional processing is required. Identity Injection policies trigger obligations.
67	No action	The rule did not initiate any action.
68	Condition false	The condition evaluated to False.
69	Condition true	The condition evaluated to True.
70	Condition unknown	Condition input was not available, so the results are unknown.
71	Cancel	The current operation has been canceled.
72	Error: Interface unavailable	The current operation is unavailable.
73	Error: Data unavailable	The data required for evaluation was unavailable.
74	Error: Illegal state	Processing error; report it to Novell® Support.

39.2.3 Role Assignment Traces

- "When the User Is Assigned Roles" on page 608
- "When the Role Policy Is Not Enabled" on page 609
- "When an Authorization Policy Uses a Role" on page 609

When the User Is Assigned Roles

Roles are assigned at authentication, so this type of trace is found in the `catalina.out` file of the Identity Server. This is a trace of a user who does not match the requirements to be assigned the Manager Role (for a definition of this Role policy, see Figure 39-1 on page 601).

```
<amLogEntry> 2007-06-11T15:38:38Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#0CE611AAE4D0301F26DD4865476BDA1
4: IDP RolesPep.evaluate(), policy trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~False(68)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~False(68)
</amLogEntry>
```

This trace describes the following about the policy.

1. The RL trace indicates that the policy has one rule and that the policy evaluated without error.
2. The RU trace indicates that the rule (`RuleID_1181251958207`) has one condition and one action and that the rule evaluated without error.
3. The CS trace indicates that the condition set evaluated to False (the user logging in does not match the conditions of the set).
4. The CO trace indicates that the condition evaluated to False (the user logging in does not match the condition).

When troubleshooting why a user is not granted access to a resource that uses a role in its Authentication policy, the first step should be to look at the `catalina.out` file of the Identity Server and determine whether the user was assigned the role. In this trace, you can see that the user was not assigned the role. To fix this problem, you can either change the conditions of the Role policy to match the user or change the user's information so that the user matches the existing condition in the Role policy.

When the Role Policy Is Not Enabled

Sometimes a Role policy is created, but the Role policy is not enabled for the Identity Server. When this happens, the trace looks similar to the following:

```
<amLogEntry> 2007-06-11T16:06:03Z INFO NIDS Application: AM#500199050:
AMDEVICEID#9921459858EAAC29: AMAUTHID#FDE680ABE320B682038947EA5F59D6B
F: IDP RolesPep.evaluate(), policy trace:
   ~~RL~0~~~~Rule Count: 0~~Success(67)
 </amLogEntry>
```

When you see Role policy traces that contain only the RL trace line, you need to enable the Role policy.

When an Authorization Policy Uses a Role

When a user requests access to a resource that has an Authorization policy that uses a role, the user is checked for the role assignment. The trace of this evaluation is in the embedded service provider log file of the Access Gateway that is processing the request. Such a trace looks similar to the following:

```
<amLogEntry> 2007-07-13T22:13:29Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-51A474B83BFDDF4F: AMAUTHID#4538DB6F6E2A237FDE674F0C6E1
6DCEC: PolicyID#N748097P-3507-3KP7-4241-410PN4152094: NXPESID#1718:
AGAuthorization Policy Trace:
   ~~RL~1~~~~Rule Count: 1~~Success(0)
   ~~RU~RuleID_1182876316974~Allow_Sales~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~NOT~1~~True(69)
   ~~CO~1~CurrentRoles(6660):no-param:authenticated~com.novell.nxpe.
condition.NxpeOperator@string-substring~SelectedRole(6661):hidden-
param:hidden-value:~~~False(68)
   ~~PA~1~~Deny Access Messasge~Sorry, you must work in sales
today.~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherCon
tainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManag
erContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Allow_Sales)
,Rule=(1::RuleID_1182876316974),Action=(Deny::1)~~~~Success(0)
</amLogEntry>
```

This trace is for a Deny policy that denies access if the user has not been assigned the Sales role. The CO line indicates that the condition is looking for a role and that the user did not match the condition.

The CS line indicates that the condition is a negative condition, meaning that the user matches the condition set when the user does not match the condition. This is the case for this user, so the condition set evaluates to True, and the action is then applied.

The PA line describes the action that was applied.

39.2.4 Identity Injection Traces

The following traces explain what to look for in an Identity Injection policy that injects an authorization header:

- "When the User Is Authenticated" on page 610
- "When the User Hasn't Authenticated" on page 611

When the User Is Authenticated

The following trace is for an Identity Injection policy that successfully inserts an authentication header. The policy inserts LDAP credentials for the user's name and password. The Access Gateway injects the information, so the trace for this type of policy is in the embedded service provider log file of the Access Gateway.

```
<amLogEntry> 2007-06-11T19:02:44Z INFO NIDS Application: AM#501103050:
AMDEVICEID#esp-534FD0D0E32FE4BD: AMAUTHID#61D5D5B3FF98156F8E4F2875981D
4A6E: PolicyID#51N4214K-74L1-491L-7190-2M9K04K21393: NXPESID#726:
AGIdentityInjection Policy Trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251426062~basic_auth_ii~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~uid~uid(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40
~40~40~40~2Fcp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredential
s~22~5D~2Fcp~3AEntry~5Bcp~3AName~3D~22UserName~22~5D:~Ok~Success(0)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~password~pwd(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40
~40~40~40~2Fcp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredential
s~22~5D~2Fcp~3AEntry~5Bcp~3AName~3D~22UserPassword~22~5D:~Ok~Success
(0)
   ~~PC~ActionID_1181251427701~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VC
DN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc
),Policy=(basic_auth_ii),Rule=(1::RuleID_1181251426062),Action=(Inject
AuthHeader::ActionID_1181251427701)~~~~Success(0)
 </amLogEntry>

<amLogEntry> 2007-06-11T19:02:44Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-534FD0D0E32FE4BD: AMAUTHID#61D5D5B3FF98156F8E4F2875981D
4A6E: PolicyID#51N4214K-74L1-491L-7190-2M9K04K21393: NXPESID#726:
Response sent: Status - success </amLogEntry>
```

Each identity injection policy generates two log entries. The first entry indicates whether the policy could successfully retrieve the information and inject it into the header. The second entry specifies whether the response is successfully sent to the Web server. This first log entry describes the following about this policy:

1. In the correlation tags (AM... tags), notice the ID assigned to the authenticated user making the request (`AMAUTHID#61D5D5B3FF98156F8E4F2875981D4A6E`).
2. After the correlation tags, the trace specifies the ID of the policy (`51N4214K-74L1-491L-7190-2M9K04K21393`).

3. The RU trace indicates that the policy name is basic_auth_ii, that the policy has no conditions, and that the policy has one action rule.
4. The first PA trace indicates that the uid (called LDAP User Name in the UI) of the Credential Profile has been successfully retrieved.
5. The second PA trace indicates that the password of the Credential Profile has been successfully retrieved.
6. The PC trace indicates that these items have been successfully injected into the header.

You can use the user's ID and the policy ID to find log entry that traces the response to the Web server. The second log entry indicates that the response was successfully sent to the Web server.

When the User Hasn't Authenticated

If the user has not authenticated and therefore has no authentication credentials, the trace for an Identity Injection policy with an authentication header looks similar to the following:

```
<amLogEntry> 2007-06-11T20:16:51Z INFO NIDS Application: AM#501103050:
AMDEVICEID#esp-534FD0D0E32FE4BD: PolicyID#OL8659PL-0K69-0N0N-0845-
5PN113KM3842: NXPESID#2539: AGIdentityInjection Policy Trace:
   ~~RL~0~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251426062~basic_auth_ii~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1181251427701~~Inject Auth Header~uid~uid(1):
CredentialProfile(7010:):NEPXurn~3Anovell~3Acredentialprofile~3A2005-
03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40
~40~40~40~2Fcp~3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredential
s~22~5D~2Fcp~3AEntry~5Bcp~3AName~3D~22UserName~22~5D:~Ok~Success(0)
   ~~PA~ActionID_1181251427701~~Inject Auth
Header~password~pwd(1):CredentialProfile(7010:):NEPXurn~3Anovell~3Acre
dentialprofile~3A2005-03~2Fcp~3ASecrets~2Fcp~3ASecret~2Fcp~3AEntry
~40~40~40~40WSCQSSToken~40~40~40~40~2Fcp~3ASecrets~2Fcp~3ASecret~5Bcp~
3AName~3D~22LDAPCredentials~22~5D~2Fcp~3AEntry~5Bcp~3AName~3D~22UserPa
ssword~22~5D:~Ok~Success(0)
   ~~PC~ActionID_1181251427701~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VC
DN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc
),Policy=(basic_auth_ii),Rule=(1::RuleID_1181251426062),Action=(Inject
AuthHeader::ActionID_1181251427701)~~~~Success(0)
 </amLogEntry>

<amLogEntry> 2007-06-11T20:16:51Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-534FD0D0E32FE4BD: PolicyID#OL8659PL-0K69-0N0N-0845-
5PN113KM3842: NXPESID#2539: Response sent: Status - success </
amLogEntry>
```

These entries look very similar to the entries for a successful injection of data. This is because injecting NULL data for data that is not available is considered a successful action. The trace displays data unavailable errors only when errors occur retrieving data. The key to determining whether the data was available for injection into an authentication header is to look for the `AMAUTHID` correlation tag in the log entry. The log entries for the `OL8659PL-0K69-0N0N-0845-5PN113KM3842` policy do not contain an `AMAUTHID` correlation tag, which indicates that the user is not logged in.

39.2.5 Authorization Traces

Authorization policies for a protected resource might require a user to be authenticated before the data required by the policy can be obtained, but Authorization policies can be configured to use data that is available without authentication. The following traces show how the log entries for an Authorization policy trace are slightly different when the user is not authenticated.

- "When the Protected Resource Requires Authentication" on page 612
- "When the Protected Resource Does Not Require Authentication" on page 613

For a trace of an Authorization policy that uses a role, see "When an Authorization Policy Uses a Role" on page 609.

When the Protected Resource Requires Authentication

The following is a successful trace of an Authorization policy that requires the user to have the value of Manager in an LDAP attribute, title. To obtain this data, the user must be authenticated.

The policy contains two rules: a Permit rule if the user has the value of Manager in the title attribute, and a Deny rule that denies all other users. This policy has been assigned to protect an Access Gateway resource.

```
<amLogEntry> 2007-08-02T15:55:05Z INFO NIDS Application: AM#501101050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#459O8443-N8P5-KO21-68OM-
K172P107N4O5: NXPESID#1743: Evaluating policy </amLogEntry>

<amLogEntry> 2007-08-02T15:55:06Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#838976482579AF372C31C4727
4E9CB28: PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: NXPESID#1743:
AGAuthorization Policy Trace:
   ~~RL~1~~~~Rule Count: 2~~Success(0)
   ~~RU~RuleID_1186068489688~Title_auth~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapAttribute(6647):NEPXurn~3Anovell~3Aldap~3A2006-
02~2Fldap~3AUserAttribute~40~40~40~40WSCQLDAPToken~40~40~40~40~2FUserA
ttribute~5B~40ldap~3AtargetAttribute~3D~22title~22~5D:hidden-
value:~com.novell.nxpe.condition.NxpeOperator@string-
equals~(0):hidden-param:hidden-value:~~~True(69)
   ~~PA~1~~Permit Access~~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisher
Container,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessMa
nagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(Title_aut
h),Rule=(1::RuleID_1186068489688),Action=(Permit::1)~~~~Success(0)
</amLogEntry>

<amLogEntry> 2007-08-02T15:55:06Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-2FA73CE1A376FD91: AMAUTHID#838976482579AF372C31C47274E
9CB28: PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: NXPESID#1743:
Response sent: Status - success </amLogEntry>
```

The first log entry is the request to evaluate the policy. The second log entry is the evaluation of the policy. The third log entry is the response that is returned. These three log entries can be tied together by using the following tags:

AMDEVICEID#esp-2FA73CE1A376FD91: When a policy evaluation request is made, the same embedded service provider processes the request. Even if the Access Gateways are clustered, the policy evaluation request stays with the Access Gateway that initiated the request.

PolicyID#459O8443-N8P5-KO21-68OM-K172P107N4O5: Each policy is assigned a unique ID, and this is the ID assigned to the policy called Title_auth in the Administration Console. To search for all log entries for a policy, use the policy ID. To search for log entries that evaluate the policy, use the policy name.

AMAUTHID#838976482579AF372C31C47274E9CB28: The request to evaluate a policy does not contain the ID of the user the request is being made for, but the log entries for the evaluation and the for the response status always contain the ID of an authenticated user. If the policy can be evaluated without the user being authenticated, these entries do not contain the ID of the user. This kind of policy might be assigned to a public resource (no authentication required) and use the time of day condition or day of the week condition for its evaluation criteria. See “When the Protected Resource Does Not Require Authentication” on page 613.

When the Protected Resource Does Not Require Authentication

The following trace is for an Authorization policy that uses data that is available without authentication. Authorization policies support a number of these conditions, such as Current Date, Current Day of Week, Current Day of Month, Current Time Of Day, Client IP, and the URL conditions. As long as you do not select to compare what is currently in the HTTP request with a value that requires authentication (such as LDAP attribute), the Authorization policy can be evaluated for an unauthenticated user. The following trace is for a policy with a Current Time of Day condition. The protected resource does not require authentication, so everyone can access the resource if their request comes in between 8:00 am and 5:30 pm, local time.

```
<amLogEntry> 2007-08-03T16:30:48Z INFO NIDS Application: AM#501101050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-
L58L08MN4N5M: NXPESID#4515: Evaluating policy </amLogEntry>

<amLogEntry> 2007-08-03T16:30:48Z INFO NIDS Application: AM#501102050:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-
L58L08MN4N5M: NXPESID#4515: AGAuthorization Policy Trace:
   ~~RL~1~~~~Rule Count: 2~~Success(0)
   ~~RU~RuleID_1186082720202~time_of_day~DNF~~1:1~~Success(0)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~0~TimeOfDay(1005):::Fri Aug 03 10:30:48 MDT
2007(9:30):~com.novell.nxpe.condition.NxpeOperator@time-in-
range~(0)::::~~~True(69)
   ~~PA~1~~Permit Access~~~~Success(0)
   ~~PC~1~~Document=(ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherCon
tainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessManag
erContainer,o=novell:romaContentCollectionXMLDoc),Policy=(time_of_day)
,Rule=(1::RuleID_1186082720202),Action=(Permit::1)~~~~Success(0)
 </amLogEntry>

<amLogEntry> 2007-08-03T16:30:48Z INFO NIDS Application: AM#501101021:
AMDEVICEID#esp-2FA73CE1A376FD91: PolicyID#216660PM-429P-O660-N25N-
L58L08MN4N5M: NXPESID#4515: Response sent: Status - success </
amLogEntry>
```

The first log entry is the request to evaluate the policy. The second log entry is the evaluation of the policy, and from it you can tell that the user is not authenticated because the `AMAUTHID#` tag is missing. The third log entry is the response that is returned, and it indicates that a success was returned. The user is allowed access to the resource.

39.2.6 Form Fill Traces

The following sections describe how to enable logging for Form Fill policies, describe the form that was used to create the Form Fill trace, then describe the entries that can be found in the logs:

- "Enabling Form Fill Logging" on page 614
- "Sample Form and Policy Used for the Trace" on page 615
- "Embedded Service Provider Trace" on page 617
- "Proxy Service Trace" on page 618

Enabling Form Fill Logging

Two modules evaluate the Form Fill policy and log entries:

- The embedded service provider of the Access Gateway evaluates the Form Fill policy and logs entries to its file. For the Linux Access Gateway, the embedded service provider sends the messages to the `catalina.out` file. For the NetWare Access Gateway, the embedded service provider sends the messages to the system console. To enable embedded service provider logging, see Section 39.1, "Turning on Logging for Policy Evaluation," on page 599.
- The proxy service of the Access Gateway reports on the process of finding the form data and filling it in. Each platform uses a different method to enable verbose logging of this process.

Linux Access Gateway: For verbose entries on a Form Fill policy, you need to use the SOAP messages logged to the `/var/log/lagsoapmessages` file. For more information, see "Configuring Logging of SOAP Messages and HTTP Headers" on page 672.

NetWare Access Gateway: On the NetWare® Access Gateway, you can add a command to the startup NCF file that increases the detail generated from a Form Fill event.

Edit the `sys:system\ap_start.ncf` file, and add the following to the `load sso` line.

```
load sso /D<number> L<number>
```

Replace <*number*> with a value of 1 to 5, with 5 specifying the most detail.

Table 39-8 describes these options:

Table 39-8 *Logging Options*

Switch	Purpose
/D<number>	Enables Form Fill debugging at the specified level: ◆ 0 enables standard output ◆ 1-5 enables debugging output, with each higher level providing more information. Default: 0

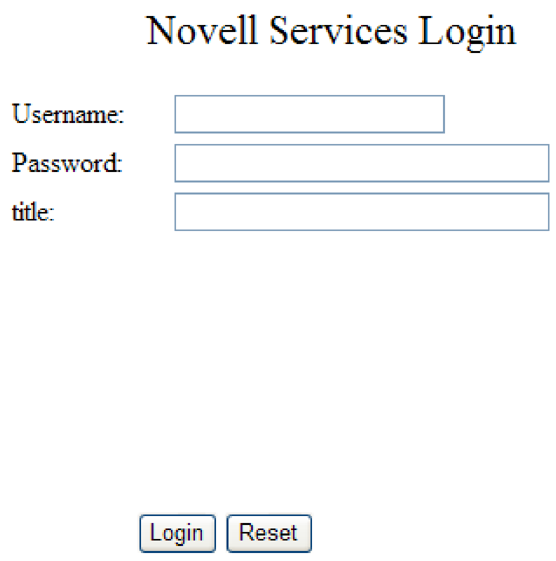
Switch	Purpose
/L<number>	Enables or disables the logging of Form Fill debugging information. ◆ 0 disables logging of Form Fill debug information ◆ 1 enables logging to the Logger screen ◆ 2 enables logging to the Extended log file ◆ 3 enables logging to both the Logger screen and the Extended log file Default: 0

For this setting to take effect, you need to restart the NetWare Access Gateway.

Sample Form and Policy Used for the Trace

Figure 39-2 illustrates the simple form that was used for the trace.

Figure 39-2 *Form Used for the Trace*



Source HTML for the Form

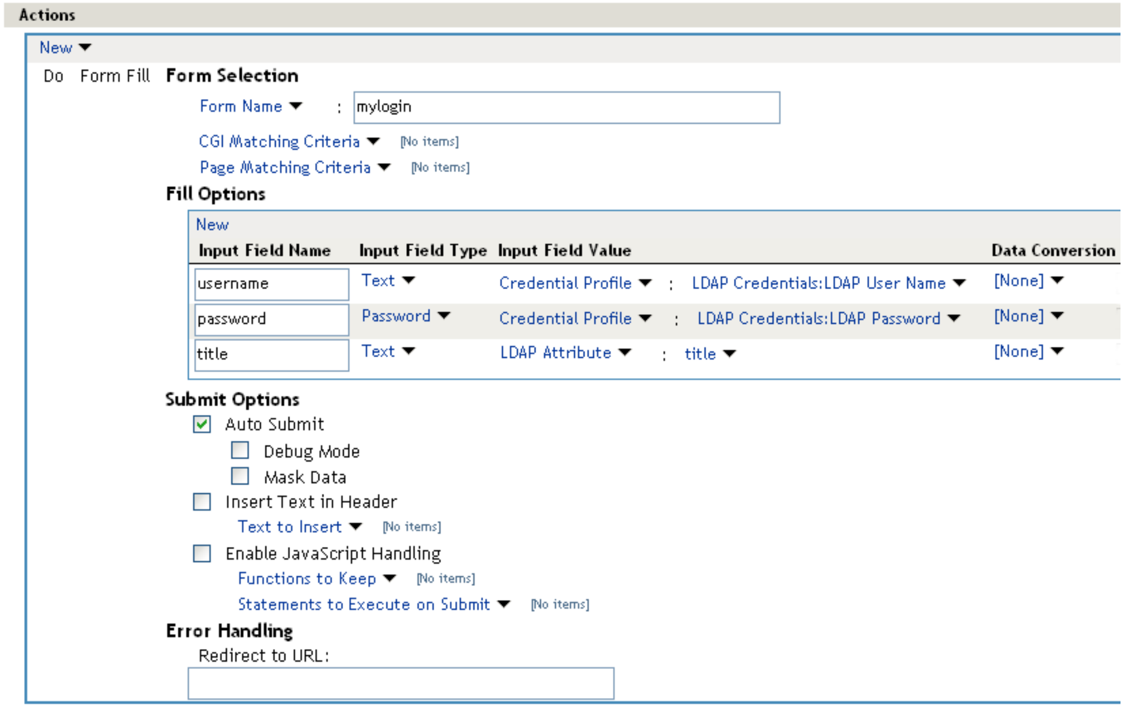
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=utf-8">
  <title>kelly</title>
</head>
<body>
  <form name="mylogin" action="double.php" method="post" id="mylogin">
    <center>
      <table border="0" cellpadding="4" cellspacing="4" width="570">
        <tr>
          <td width="121" height="285" align="left" valign="top">
          </td>
          <td width="449" height="285" align="center" valign="top">
            <p align="center">
              <font size="5">Novell Services Login<br></font>
            </p>
            <table border="0" width="86%">
              <tr>
                <td width="25%">Username:</td>
                <td width="75%">
                  <input type="TEXT" name="username">
                </td>
              </tr>
              <tr>
                <td width="25%">Password:</td>
                <td width="75%">
                  <input type="PASSWORD" name="password" size="30">
                </td>
              </tr>
              <tr>
                <td width="25%">title:</td>
                <td width="75%">
                  <input type="TEXT" name="title" size="30">
                </td>
              </tr>
            </table>
          </td>
        </tr>

         <tr>
            <td colspan="2" align="center">
             <input type="hidden" name="formNum" value="1">
             <input type="submit" value="Login">
             <input type="reset">
            </td>
          </tr>
        </table>
      </center>
    </form>
</body>
</html>
```

The name of the form and the fields that need to be filled in by the policy are in bold typeface.

Form Fill Policy

The following Form Fill policy was created for the `mylogin` form. The policy is called `simpleform`. You can use the name of the policy to find entries for it in the log files. The policy was assigned to the `/identity/forms/simple.html` protected resource. Because the URL path identifies a specific file on the Web server, the policy does not require any CGI or page matching criteria.

Figure 39-3 The Form Fill Policy for the mylogin Form


Actions

New

Do Form Fill **Form Selection**

Form Name : mylogin

CGI Matching Criteria [No items]

Page Matching Criteria [No items]

Fill Options

New

Input Field Name	Input Field Type	Input Field Value		Data Conversion
username	Text	Credential Profile :	LDAP Credentials:LDAP User Name	[None]
password	Password	Credential Profile :	LDAP Credentials:LDAP Password	[None]
title	Text	LDAP Attribute :	title	[None]

Submit Options

☑ Auto Submit

☐ Debug Mode

☐ Mask Data

☐ Insert Text in Header

Text to Insert [No items]

☐ Enable JavaScript Handling

Functions to Keep [No items]

Statements to Execute on Submit [No items]

Error Handling

Redirect to URL:

This policy is configured so that the user never sees it. Even on first login, the form is filled in for authenticated users because the user's authentication credentials are used for the username and password fields, and the title field value is obtained from the LDAP user store. If the user does not have a value for the title attribute, the user sees the form every time the page is accessed. If you want the value to be saved for these users, you need to change the policy to use a Secret Store rather than an LDAP attribute.

Embedded Service Provider Trace

When looking for entries for the simpleform policy in the embedded service provider trace, you can use the following strings to find the entries:

- The name of the Form Fill policy: `simpleform`
- The string identifying a Form Fill trace: `AGFormFill Policy Trace`
- The policy ID (after you have found it): `PolicyID#06O0287L-06LO-KKP4-207M-6971PPM6147L`

The following trace is from the `catalina.out` file of the embedded service provider of a Linux Access Gateway. The entries have been number so that they can be described, and a few extra line breaks and spaces have been added to make the entries easier to read.

```
1. <amLogEntry> 2007-09-14T00:15:52Z INFO NIDS Application:
AM#501101050: AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06O0287L-06LO-
KKP4-207M-6971PPM6147L: NXPESID#2663: Evaluating policy </amLogEntry>

2. <amLogEntry> 2007-09-14T00:15:52Z INFO NIDS Application:
AM#501104050: AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06O0287L-06LO-
KKP4-207M-6971PPM6147L: NXPESID#2663: AGFormFill Policy Trace:
   ~~RL~1~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1189711482510~simpleform~DNF~~0:1~~Success(67)
   ~~PA~ActionID_1189711485006~~Added Form Selection Group~~~~Success
       (0)
   ~~PA~ActionID_1189711485006~~Added Fill Options Group~~~~Success(0)
   ~~PA~ActionID_1189711485006~~Added Submit Options Group~~~~Success
       (0)
   ~~PC~ActionID_1189711485006~~Document=(ou=xpemlPEP,ou=mastercdn,
     ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,
     ou=VCDN_Root,ou=accessManagerContainer,o=novell:romaContent
     CollectionXMLDoc),Policy=(simpleform),Rule=(1::RuleID_11897114
     82510),Action=(FormFill::ActionID_1189711485006)~~~~Success(0)
 </amLogEntry>

3. <amLogEntry> 2007-09-14T00:15:52Z INFO NIDS Application:
AM#501101021: AMDEVICEID#esp-917A1174C8A270FC: PolicyID#06O0287L-06LO-
KKP4-207M-6971PPM6147L: NXPESID#2663: Response sent: Status - success
</amLogEntry>
```

1. The first log entry is the request to evaluate the policy. If this entry doesn't occur, make sure that the Form Fill policy is enabled for the protected resource.
2. The second entry is the actual policy trace. For a Form Fill policy, it is fairly basic information about the three types of actions in the policy: matching the form, filling in the field options, and adding the submit options. To determine what information was put in the options, you need to view the proxy service trace.
3. The third entry indicates the type of response that is returned from the evaluation. In this entry, success is returned.

Proxy Service Trace

When looking for entries in the proxy trace of the Access Gateway log, you can use the following strings to find the entries:

- The event code of a Form Fill event: `AM#504507000`
- The name of the Form Fill policy: `simpleform`
- The name of the form: `mylogin`
- The names of the fill option fields: `username, password, title`

The sample trace is from a `ics_dyn.log` file of a Linux Access Gateway. Some of the lines are very long, and extra white space has been added to make them easier to read. The first occurrence of an item you can search for is displayed in a bold typeface.

```
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
AMEVENTID#0: ***************************************************
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Name : (mastercdnsimpleform3310)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Type : (FILL)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: CGI Matching Criteria: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
```

```
  AMEVENTID#0: Page Matching Criteria:
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Not Configured.
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Form Number : (-1)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Form Name: (mylogin)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Form Id: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Login Fail Redirect: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Login Fail Delete Rem: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Error Redirect: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Post options (silent = yes), (debug = no), (masked =
  no), (enabled = yes)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: InsertText: ()
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: JavaScriptHandling:
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Not configured.
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Fill Option 0 : ( Name=username, Value=NEPXurn~
  3Anovell~3Acredentialprofile~3A2005-03~2Fcp~3ASecrets~2Fcp~3ASecret~
  2Fcp~3AEntry~40~40~40~40WSCQSSToken~40~40~40~40~2Fcp~3ASecrets~
  2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredentials~22~5D~2Fcp~3AEntry~
  5Bcp~3AName~3D~22UserName~22~5D, DataConversion=None,
  valType=CREDENTIAL_PROFILE, inputType=TEXT, isDuplicate=false)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
  AMEVENTID#0: Fill Option 1 : ( Name=password, Value=NEPXurn~
  3Anovell~3Acredentialprofile~3A2005-03~2Fcp~3ASecrets~2Fcp~
  3ASecret~2Fcp~3AEntry~40~40~40~40WSCQSSToken~40~40~40~40~2Fcp~
  3ASecrets~2Fcp~3ASecret~5Bcp~3AName~3D~22LDAPCredentials~22~5D
  ~2Fcp~3AEntry~5Bcp~3AName~3D~22UserPassword~22~5D,
  DataConversion=None, valType=CREDENTIAL_PROFILE, inputType=PASSWORD,
  isDuplicate=false)
Sep 19 08:04:50 jwilson : AM#504507000: AMDEVICEID#ag-: AMAUTHID#0:
   AMEVENTID#0: Fill Option 2 : ( Name=title, Value=NEPXurn~
   3Anovell~3Aldap~3A2006-02~2Fldap~3AUserAttribute~40~40~40~
   40WSCQLDAPToken~40~40~40~40~2FUserAttribute~5B~40ldap~
   3AtargetAttribute~3D~22title~22~5D, DataConversion=None,
   valType=LDAP_ATTRIBUTE, inputType=TEXT, isDuplicate=false)
```

On the Linux Access Gateway, you can get more detailed information on the process that was used to fill the form when you turn on logging to the `lagsoapmessages` file. For more information, see “Configuring Logging of SOAP Messages and HTTP Headers” on page 672.

39.3 Common Configuration Problems That Prevent a Policy from Being Applied as Expected

When trying to determine what is functioning incorrectly in a policy, you need to turn on policy tracing and understand the evaluation traces. See the following:

- Section 39.1, "Turning on Logging for Policy Evaluation," on page 599
- Section 39.2, "Understanding Policy Evaluation Traces," on page 600

The CO entry line of a policy trace identifies when a policy condition evaluates to False or True. The PA entry line indicates whether the Action was applied or ignored. If the results of the policy trace are not what you expected for the user, the next step is to determine why the policy isn't behaving the way you want it to. Check for the following problems:

- Section 39.3.1, "Enabling Roles for Authorization Policies," on page 620
- Section 39.3.2, "LDAP Attribute Condition," on page 620
- Section 39.3.3, "Result on Condition Error Value," on page 621
- Section 39.3.4, "An External Secret Store and Form Fill," on page 622

39.3.1 Enabling Roles for Authorization Policies

If you are using roles in your authorization policies, you need to make sure that the role is enabled for the Identity Server configuration. You can create roles and authorization policies independently of assigning them to protect a resource or to an Identity Server configuration.

If you haven't enabled the role, users are not assigned the role when they log in, even when they meet all the criteria for the role.

- If the Authorization Policy is an Allow policy, the users might be denied access because they haven't been assigned the role.
- If the Authorization Policy is a Deny policy, the users might be allowed access because they haven't been assigned the role.

Whenever an Authorization Policy is not producing the expected results and the policy contains a role, the first troubleshooting step should always be to check whether the role has been enabled for the Identity Server configuration. Click *Access Manager* > *Identity Servers* > *Edit* > *Roles*. If the role is not enabled, the Identity Server cannot assign the role to the user.

The second step should be to ensure that the roles are transferred from for Identity Server to the embedded service provider. Click *Access Manager* > *Identity Servers* > *Edit* > *Liberty* > *Web Service Provider*. The *Authentication Profile* needs to be enabled in order for embedded service providers to evaluate roles in policies. This profile is enabled by default, but it can be disabled. When disabled, all devices assigned to use this Identity Server cluster configuration cannot determine which roles a user has been assigned, and the devices evaluate policies as if the user has no roles.

39.3.2 LDAP Attribute Condition

If you use an LDAP attribute as the condition for a Role policy or an Authorization policy and your users are not being assigned the role or allowed (denied) access to a resource, the most likely cause of the problem is the LDAP attribute name used in the policy. Some administration tools for the

LDAP user stores display a UI name or an eDirectory name rather than the LDAP attribute name. Access Manager policies require the LDAP attribute name.

Use the following steps to identity whether the Access Manager policy has been configured for the LDAP attribute name, a UI name, or an eDirectory name:

1. Use an LDAP browser to view one of your users in your LDAP user store.

   You can download a Java-based tool from LDAP Browser/Editor (http://www-unix.mcs.anl.gov/~gawor/ldap/).
2. Verify the LDAP name of the attribute and that the user has the expected value.
3. In the Administration Console, click *Access Manager* > *Policies* > *[Name of Policy]* > *Rule Number*.
4. View the attribute name and value for the LDAP Attribute condition.
5. Verify the following:
   - The name of the attribute should match the name as displayed in the LDAP browser. The attribute name is not case sensitive, but it should not contain any spaces. If you need to modify the attribute used by the policy, click the attribute name, then select one from the list or select *New LDAP Attribute* to add one.
   - The value can be case sensitive, depending upon how you have configured the *Mode* for the policy. If you have selected case sensitive for the *Mode*, make sure the case in the policy matches the case in the LDAP user store.
   - If the attribute is multi-valued and your users typically have multiple values, select *Substring* as the *Comparison* type.
6. If these steps have not solved the problem, see Section 39.3.3, "Result on Condition Error Value," on page 621.

39.3.3 Result on Condition Error Value

If you incorrectly set the value of the *Result on Condition Error* field, you create a policy that allows an action that you want the policy to deny or that denies an action that you want allowed. You must carefully evaluate whether you want the action applied or ignored when an error occurs during the evaluation of the condition. For positive conditions, the following rules apply:

- For the action to be applied, either the user must match the condition or the *Result on Condition Error* must be set to True.
- For the action to be ignored, either the user must not match the condition or the *Result on Condition Error* must be set to False.

The logic is harder to follow when you start adding "if not" to the conditions. The user then matches the condition by not matching the condition. For this type of condition, you need to ask whether you want the action applied to any user when an error occurs evaluating the condition.

The logic is even harder to following when you start adding multiple condition groups that can also have "or nots" and "if nots".

If you have a policy that uses "if not" conditions or uses multiple condition groups and it is not producing the expected results, you might want to rewrite the policy so that it contains only positive conditions. You might want to modify the condition groups so that the policy uses multiple rules, with each rule containing one condition group with the conditions you want the user to match for the action you assign to the rule.

39.3.4 An External Secret Store and Form Fill

When you create a user store on the Identity Server (*Local* > *User Stores*) and define it as an external Secret Store (*Liberty* > *Web Service Provider* > *Credential Profile*), some attributes are not being created properly on the SAML affiliate object. The workaround is to access the user store configuration page (*Local* > *User Stores*), then exit. This action results in a check to verify that the schema, objects, and attributes exist, and recreates the affiliate object from scratch, if necessary.

The following affiliate objects must exist:

```
authsamlCertContainerDN (container holding trusted certificates,
   for example: SCC Trusted Root.Security)
authsamlProviderID
authsamlTrustedCertDN (list of trusted certificate(s))
authsamlValidAfter (180 seconds default)
authsamlValidBefore (180 seconds default)
```

If these attributes exist, the system works normally. However, your Identity Server and Secret Store server are not synchronized for time. If time sync is an issue, you can change the 180 second default validity times as a workaround.

If your LDAP user store and the Administration Console have a firewall separating them, TCP ports 524 and 636 must be open to allow for the creation of the required objects. For more information about ports and firewalls, see "Setting Up Firewalls" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

39.4 The Policy Seems to Be Using Old User Data

When a policy is first evaluated, it caches information about the user. Some data items are updated every minute. Some are cached for the duration of the request. Some are cached for the duration of the user's session. When a data item is cached for the duration of a user session, the user must log out and log in for the policy modification to take effect.

Table 39-9 lists how long the data items for a condition are cached before being refreshed.

Table 39-9 *Data Caching Limits*

Condition	Data Refresh Interval
Authenticating IDP	User session
Authentication Contract	User session
Authentication Method	User session
Authentication Type	User session
Client IP	Request
Credential Profile	User session

Condition	Data Refresh Interval
Current Date	One minute
Current Day of Week	One minute
Current Day of Month	One minute
Current Time of Day	One minute
HTTP Request Method	Request
Java Data Injection Module	User session
LDAP Attribute	User session
LDAP Group	User session
LDAP OU	User session
Liberty User Profile	User session
Proxy Session Cookie	User session
Roles for Current User	User session
Roles from Identity Provider	User session
Shared Secret	Request
String Constant	User session
URL	Request
URL Scheme	Request
URL Host	Request
URL Path	Request
URL File Name	Request
URL File Extension	Request
User Store	User session
X-Forward-For IP	Request

39.5 Checking for Corrupted Policies

For a policy to be evaluated correctly, the policy must contain a rule. To verify that your system does not contain any policies with configuration errors:

1. In the Administration Console, click *Access Manager* > *Auditing* > *Troubleshooting* > *Policies*.

   If you have any corrupted policies, they appear in the list.
2. Identify the corrupted policy, then click *Remove*.

39.6 Policy Page Timeout

If your policy page hangs, and you have an LDAP group or LDAP ou being used in the policy, check the health of your user stores (LDAP servers) and ensure that they are communicating.

39.7 Policy Creation and Storage

For troubleshooting, you can export the policy and send it to Novell for debugging. If the policy uses roles, make sure you also export the Role policies.

Policies are stored as XML documents in the object directory, with one XML document to represent each policy container. The default policy container (Master_Container) resides at:

```
\\novell\accessManagerContainer\VCDN_Root\PartitionsContainer\Partitio
n\ContentPublisherContainer\mastercdn\xpemlPEP\romaContentCollectionXM
LDoc
```

Other policy containers are stored following the same path, with a unique name string representing the policy name that replaces the ou=mastercdn portion of the above path.

If you are unsure if the policy is being created correctly or if you need to check to see if the policy is enabled, you can view the policy list in the interface. If you think the GUI is not properly displaying the policy, you can also view the XML by navigating to the Policy Conditions on which you edit rules, right click and choose *This Frame > View Frame Source*.

39.8 Policy Distribution

Policy definitions are not replicated, but are referenced by the Access Gateways for which the policy is to be evaluated. The policy reference mechanism is a set of XML elements that refer back to the policy definitions stored in the various policy containers. If you have configured a policy for a protected resource and an Access Gateway does not seem to be executing this policy, use the following procedures to verify that the Access Gateway has been configured to use the policy:

1. Set the level of Application logging to *config*. See Section 39.1, “Turning on Logging for Policy Evaluation,” on page 599.

   This enables the tracing of the policy enforcement lists.
2. Search for name of your policy in a `<PolicyEnforcementList>` element. The ExternalElementRef attribute contains a reference to the policy name.
   - On the Linux Access Gateway, you can find these elements in the `catalina.out` file.
   - On the NetWare Access Gateway, the trace for these elements goes to the system console.

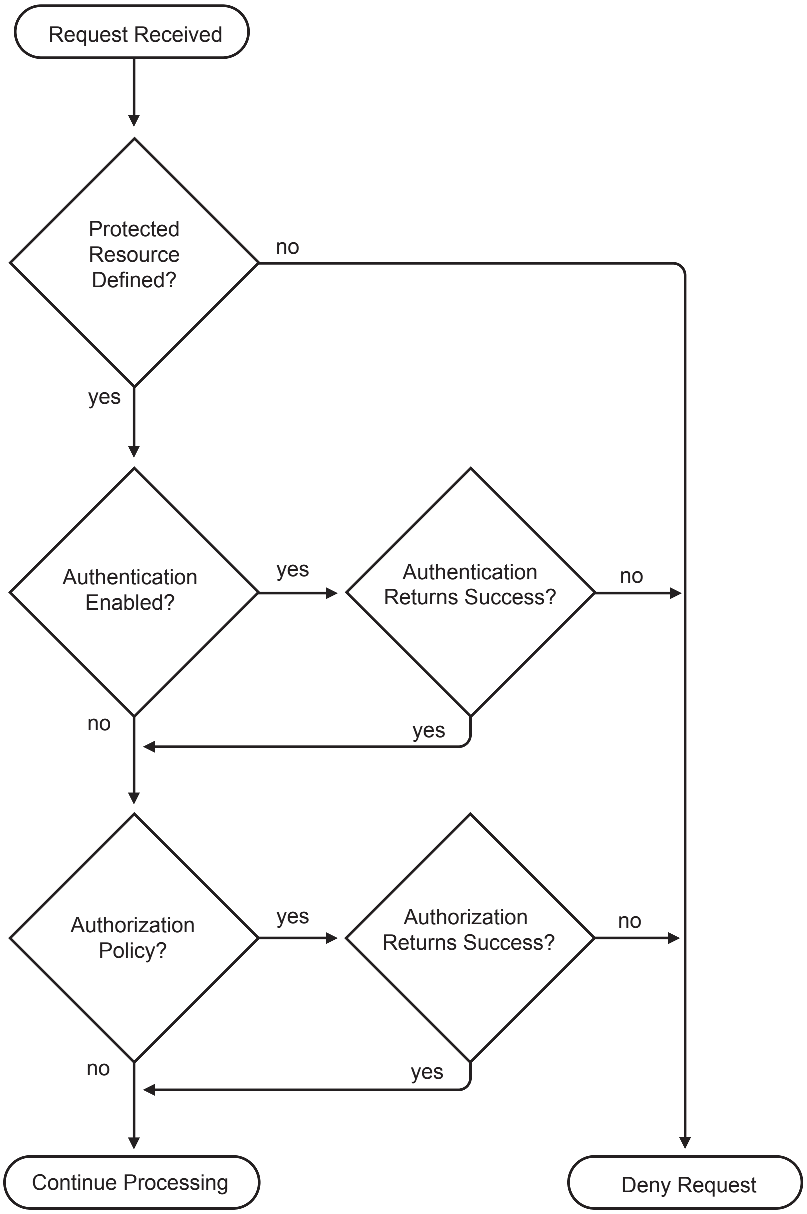
     You can also find an XML file named after each protected resource in the `sys:\etc\proxy\pr` directory. These files contain the references to the policy names that have been enabled for the protected resources.
3. If you cannot find the policy name, the Access Gateway has not been configured to use the policy. The configuration either needs to be applied or the policy needs to be enabled. For information on how to assign a policy to a protected resource, see Section 13.4, “Configuring Protected Resources,” on page 207.
4. If you find the policy name associated with the correct protected resource, you need to check why the policy is not evaluating according to your design. Set the level of Application logging

to *info* and examine the policy trace from a user accessing the protected resource. See Section 39.2, “Understanding Policy Evaluation Traces,” on page 600.

39.9 Policy Evaluation: Access Gateway Devices

The following diagram depicts how Authorization policies fit into the protected resource processing for the proxy.

Figure 39-4 *Policy Evaluation*



Policies for the Access Gateway devices are evaluated by the policy engine in Java. A SOAP interface is used to transition from the proxy to Java and back. To see the SOAP messages, you need to set the logging level of the *Application* level to *config*. See Section 39.1, “Turning on Logging for Policy Evaluation,” on page 599.

For NetWare, the SOAP messages are output to the Logger Screen. For Linux, the SOAP messages are output to the `catalina.out` file. Sample SOAP messages are shown in the following scenarios:

- Section 39.9.1, "Successful Policy Configuration Example," on page 626
- Section 39.9.2, "No Policy Defined Configuration Example," on page 627
- Section 39.9.3, "Deny Access Configuration/Evaluation Example," on page 627

39.9.1 Successful Policy Configuration Example

Note the Policy Enforcement Point (PEP) identifier of AGIdentityInjection in the request and the PolicyID in the response.

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
   envelope/">
<SOAP-ENV:Body>
   <NXPES ID="12">
      <Configure-ag PEPName="AGIdentityInjection">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138389868885"
            LastModifiedBy="cn=admin,o=novell">
            <PolicyRef ElementRefType="ExternalWithIDRef"
                ExternalElementRef="PolicyID_xpemlPEP_AGIdentity
                    Injection_ii_test"
                ExternalDocRef="ou=xpemlPEP,ou=mastercdn,
                    ou=ContentPublisherContainer,ou=Partition,
                    ou=PartitionsContainer,ou=VCDN_Root,ou=access
                    ManagerContainer,o=novell:romaContentCollection
                    XMLDoc"
                UserInterfaceID="PolicyID_xpemlPEP_AGIdentity
                    Injection_ii_test"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <ConfigureResponse PolicyId="755OK8P0-7543-518M-8L8M-N0P2LM2
                N3O27">
         <ContextDataElement Enum="2551"/>
      </ConfigureResponse>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

39.9.2 No Policy Defined Configuration Example

The following is a sample of a configuration request where the policy code detects that no policies are in effect for the protected resource and Policy Enforcement Point (PEP).

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
<SOAP-ENV:Body>
   <NXPES ID="11">
      <Configure-ag PEPName="AGAuthorization">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138389868885"
            LastModifiedBy="cn=admin,o=novell">
           <PolicyRef ElementRefType="ExternalWithIDRef"
               ExternalElementRef="PolicyID_xpemlPEP_AGIdentity
                        Injection_ii_test"
               ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
                       PublisherContainer,ou=Partition,ou=Partitions
                       Container,ou=VCDN_Root,ou=accessManager
                       Container,o=novell:romaContentCollectionXMLDoc"
               UserInterfaceID="PolicyID_xpemlPEP_AGIdentityInjection_
                       ii_test"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
      envelope/">
   <SOAP-ENV:Body>
      <NXPES Id="" Status="emptypolicyset"/>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

39.9.3 Deny Access Configuration/Evaluation Example

The following is a sample of a configuration request for a Deny policy and an evaluation request for this policy.

Configuration Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
   envelope/">
<SOAP-ENV:Body>
   <NXPES ID="17">
      <Configure-ag PEPName="AGAuthorization">
         <PolicyEnforcementList
            RuleCombiningAlgorithm="DenyOverridesWithPriority"
            schemaVersion="1.32"
            LastModified="1138718667305"
            LastModifiedBy="cn=admin,o=novell">
         <PolicyRef
            ElementRefType="ExternalWithIDRef"
            ExternalElementRef="PolicyID_xpemlPEP_AGIdentityInjection
                _custom_test"
            ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
               PublisherContainer,ou=Partition,ou=PartitionsContainer,
               ou=VCDN_Root,ou=accessManagerContainer,o=novell:roma
               ContentCollectionXMLDoc"
            UserInterfaceID="PolicyID_xpemlPEP_AGIdentityInjection
               _custom_test"/>
         <PolicyRef
            ElementRefType="ExternalWithIDRef"
            ExternalElementRef="PolicyID_xpemlPEP_AGAuthorization_
               deny-all"
            ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=Content
               PublisherContainer,ou=Partition,ou=PartitionsContainer,
               ou=VCDN_Root,ou=accessManagerContainer,o=novell:roma
               ContentCollectionXMLDoc"
            UserInterfaceID="PolicyID_xpemlPEP_AGAuthorization
               _deny-all"/>
         </PolicyEnforcementList>
      </Configure-ag>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Configuration Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
    envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <ConfigureResponse
           PolicyId="55N3NL81-L29N-2619-K0M8-2L963M0MM701"/>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Evaluation Request

```
toBufSeg: <?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/">
<SOAP-ENV:Body>
   <NXPES ID="18">
      <Evaluate PolicyId="55N3NL81-L29N-2619-K0M8-2L963M0MM701"
                Verbose="on"/>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Evaluation Response

```
LibertyProcessMsgCB:
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
      envelope/">
<SOAP-ENV:Body>
   <NXPES Id="" Status="success">
      <EvaluateResponse>
         <DoAction ActionName="Deny" ActionTTL="-1" Enum="2620">
            <Parameter Enum="10" Name="Message" Value=""/>
         </DoAction>
      </EvaluateResponse>
   </NXPES>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Troubleshooting the Access Gateway

40

For a solution to an Access Gateway problem, see the following sections:

- Section 40.1, “Fixing Problems Common to Both Platforms,” on page 631
- Section 40.2, “Troubleshooting the Linux Access Gateway,” on page 635
- Section 40.3, “Troubleshooting the NetWare Access Gateway,” on page 657

40.1 Fixing Problems Common to Both Platforms

The following problems occur on both the Linux Access Gateway and the NetWare® Access Gateway.

- Section 40.1.1, “Rewriting Fails on a Page with Numerous HREFs,” on page 631
- Section 40.1.2, “Troubleshooting HTTP 1.1 and GZIP,” on page 632
- Section 40.1.3, “Links Are Broken Because the Rewriter Sends the Request to the Wrong Proxy Service,” on page 633
- Section 40.1.4, “Protected Resources Referencing Non-Existent Policies,” on page 633
- Section 40.1.5, “Mismatched SSL Certificates in a Cluster of Access Gateways,” on page 634
- Section 40.1.6, “Protected Resource Configuration Changes Are Not Applied,” on page 634
- Section 40.1.7, “Recovering from a Hardware Failure on an Access Gateway Machine,” on page 634
- Section 40.1.8, “Viewing Configuration Information,” on page 635

For policy errors, see Chapter 39, “Troubleshooting Access Manager Policies,” on page 599.

For XML validation errors, see Chapter 43, “Troubleshooting XML Validation Errors,” on page 683.

40.1.1 Rewriting Fails on a Page with Numerous HREFs

Although the rewriting failure occurs when downloading large amounts of data from a protected Web server, it is not the size or the timeout of the page that is the issue. It is the number of links to be rewritten. The Access Gateway has a data size limit to the number of references that the rewriter can rewrite on a page.

The solution is to reduce the number of HREFs on the page that need to be rewritten. If the problem is occurring because the rewriter is rewriting HTTP to HTTPS, you can solve this problem by disabling multi-homing for the Web server and by rewriting the Web page to use relative links. This reduces the number of links that need to be rewritten.

40.1.2 Troubleshooting HTTP 1.1 and GZIP

HTTP 1.1 has the ability to deal with compressed data in either a Deflate or GZIP format. This reduces the size of data being sent across the wire. Because HTML pages are just text, they typically compress very well.

To use GZIP, you enable your Web servers to send GZIP-compressed data. Be aware that some Web servers do not respond with compressed (GZIP) data when the Access Gateway sends the Via header to the Web server. Check you Web server documentation.

When the Web server sends compressed data and the rewriter needs to process the data, the data is decompressed, rewritten, and then recompressed. When Form Fill needs to process the data, the date is decompressed and then processed. If the Access Gateway does not need to perform any rewriting of the data or if Form Fill does not need to process the data, the compressed data is sent unchanged from the Web server to the browser. This is the default behavior.

If you are having problems with the recompressed data on the browser, see the following:

- "NetWare Access Gateway" on page 632
- "Linux Access Gateway" on page 632

NetWare Access Gateway

To determine if the compression algorithm employed by the NetWare Access Gateway has a problem, you can turn it off. With recompression turned off, pages which had problems loading might now load properly. To turn off the compression algorithm, add the following command to the `load proxy` line of the `appstart.ncf` file and then restart the system:

```
load proxy -gzip 0
```

This causes the NetWare Access Gateway to accept compressed data from the Web server and to send uncompressed data to the browser.

To return the NetWare Access Gateway to the default behavior, add the following command to the `load proxy` line of the `appstart.ncf` file and then restart the system:

```
load proxy -gzip 1
```

This command causes the NetWare Access Gateway to accept compressed data from the Web server and to send recompressed data to the browser.

Linux Access Gateway

To turn off the GZIP feature:

1. Add the following touch file

   `/var/novell/.noGzipSupport`

   Use the `touch` utility to create this blank file.
2. Restart the Linux Access Gateway.

In the presence of this touch file, Linux Access Gateway does not forward the ACCEPT-ENCODING header to the Web server. Without this header, the Web server does not send any data with GZIP or Deflate encoding to the Linux Access Gateway.

To allow the Linux Access Gateway to receive GZIP or Deflate encoded data, remove the touch file and restart the Linux Access Gateway.

40.1.3 Links Are Broken Because the Rewriter Sends the Request to the Wrong Proxy Service

When links on the Web server are rewritten to the wrong proxy service, the reverse proxy and Web servers might have the following configuration:

- The initial request from the browser is to a path-based multi-homing proxy service.
- The reverse proxy is configured to service one or more path-based proxy services.
- The path-based proxy services are configured to *Forward Received Host Name* and to *Remove Path on Fill*.
- The Web servers protected by these path-based proxy services have links to each other.

With this configuration, the rewriter cannot determine whether the link is to the current proxy service, one of the other path-based proxy services, or the parent proxy service. With the path removed, all the path-based proxy services have the same name. For example if one proxy service has the published name of mycompany.provo.novell.com/sales and a second path-based proxy service has a name of mycompany.provo.novell.com/app, the names are the same as the parent proxy service when the path is removed. The HTTP header does not help, because the proxy services are forwarding the same host name: mycompany.provo.novell.com.

There are a number of ways to solve this problem. One of the easiest ways is to set up DNS names for the Web servers, then configure the proxy services so that the *Host Header* option is set to *Web Server Host Name* and the DNS name of the Web server is specified in the *Web Server Host Name* field. This places the DNS name of the Web Server name in the HTTP Host header, allowing the rewriter to distinguish it from the other Web servers protected by the reverse proxy.

40.1.4 Protected Resources Referencing Non-Existent Policies

If your protected resources contain references to policies that do not exist, use the following procedures to remove them.

1. Click *Access Manager* > *Auditing* > *Troubleshooting*.
2. In the *Access Gateways with Protected Resources Referencing Nonexistent Policies* section, click *Repair*.

   This removes the link between the protected resource and the policy.
3. Verify that correct policies are enabled on the protected resources. Click *Access Gateways* > *Edit* > *[Name of Reverse Proxy]* > *[Name of Proxy Service]* > *Protected Resources*.
4. Change to the *Policy View*.
5. (Optional) Click the *Used By* link to modify existing assignments.
6. Click *OK*, then click the *Access Gateways* link.
7. Click *Update* > *OK*.

40.1.5 Mismatched SSL Certificates in a Cluster of Access Gateways

Sometimes a newly added server in a cluster does not receive the certificate that the rest of the cluster is using for SSL. To fix this problem:

1. Click *Access Manager* > *Access Gateways* > *Edit* > *[Name of Reverse Proxy]*.
2. For the Server Certificate, click the *Select Certificate* icon, then select a different certificate, such as the test-connector certificate.
3. Click *OK* to ignore the warnings that the certificate CN does not match the reverse proxy.

   This is what you want.
4. Click *OK*.
5. Click *[Name of Reverse Proxy]*.

   This needs to be the same reverse proxy that you selected in Step 1.
6. For the Server Certificate, click the Select Certificate icon, select the certificate whose CN matches the published DNS name of the parent proxy service, then click *OK*.
7. Click *OK*.

   When you click OK, the correct certificate is added to the keystore.
8. Repeat Step 1 through Step 7 for each reverse proxy that uses a unique certificate. If all the reverse proxies use the same certificate, continue with Step 9.
9. On the Access Gateways page, click *Update* > *OK*.

   The configuration changes are pushed to the Access Gateway, and the Access Gateway loads and uses the new certificate.

40.1.6 Protected Resource Configuration Changes Are Not Applied

If you modify the configuration for a protected resource by modifying its *URL Path List*, or its Authorization, Identity Injection, or Form Fill policies, you save these changes and apply them by clicking Update, then return to the resource and the changes have not been applied, the protected resource has a corrupted configuration. To repair the configuration:

1. Click *Access Manager* > *Auditing* > *Troubleshooting*.
2. In the *Access Gateways with Corrupted Protected Resource Data* list, select the resource with the problem, then click *Repair*.

   This repairs the configuration for the selected protected resource.
3. Reconfigure the protected resource with the changes that weren’t applied.

40.1.7 Recovering from a Hardware Failure on an Access Gateway Machine

If an Access Gateway machine experiences a hardware failure, such as a failed hard disk, you can preserve its configuration and have it applied to the replacement machine. For information about this procedure, see Section 2.5, “Restoring an Access Gateway,” on page 34.

40.1.8 Viewing Configuration Information

The configuration store maintains two versions of the Access Gateway configuration and browser cache maintains one.

- **Current:** The current configuration is the version of the configuration that the Access Gateway is currently using.

  You can view this configuration in file format by clicking *Access Manager* > *Access Gateways* > *[Name of Server]* > *Configuration* > *Export*. Do not set a password to encrypt the file. The exported file contains the current configuration.
- **Working:** The working configuration is the version that you have saved by clicking the *OK* button on the Server Configuration page, but you have not applied the changes by clicking the *Update* or the *Update All* link on the Access Gateways page. This version is not viewable from the Administration Console.
- **Browser Cache:** All configuration changes are saved to browser cache when you click the *OK* button on a configuration page. To view the configuration currently in browser cache, click *Access Manager* > *Auditing* > *Troubleshooting*, scroll to the *Cached Access Gateway Configurations* section, then click *View*. You can view the cached configuration of an individual Access Gateway, or if the Access Gateway is a member of a cluster, you can view the cached configuration of the cluster and each member. The + and - buttons allow you to expand and collapse individual configurations.

40.2 Troubleshooting the Linux Access Gateway

This section provides various troubleshooting scenarios and frequently asked questions that you might encounter while using the Linux Access Gateway, and suggests appropriate actions.

- Section 40.2.1, "Useful Tools and Files for Troubleshooting the Linux Access Gateway," on page 636
- Section 40.2.2, "Troubleshooting a Failed Linux Access Gateway Configuration," on page 642
- Section 40.2.3, "Troubleshooting a Linux Access Gateway Crash," on page 642
- Section 40.2.4, "HTTP Requests Are Dropped," on page 645
- Section 40.2.5, "Linux Access Gateway Not Responding," on page 646
- Section 40.2.6, "Connection Details," on page 646
- Section 40.2.7, "Network Socket Issues," on page 646
- Section 40.2.8, "Authentication Issues," on page 647
- Section 40.2.9, "Rewriter Issues," on page 650
- Section 40.2.10, "COS Related Issues," on page 651
- Section 40.2.11, "Memory Issues," on page 653
- Section 40.2.12, "Authorization and Identity Injection Issues," on page 654
- Section 40.2.13, "Form Fill Issues," on page 655

For information about installation and import issues, see "Troubleshooting Linux Access Gateway Installation" and "Troubleshooting the Access Gateway Import" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

40.2.1 Useful Tools and Files for Troubleshooting the Linux Access Gateway

- "Useful Tools" on page 636
- "The Linux Access Gateway Console" on page 637
- "Useful Troubleshooting Files" on page 639

Useful Tools

Table 40-1 describes some of the tools available in the Linux operating system or installed by the Linux Access Gateway that can help you determine the cause of a problem.

Table 40-1 *Useful Tools*

Tool	Description
Health icon	In the Administration Console, click on the *Health* icon to view details about the health of the Access Gateway. For more information, see Section 34.3, "Monitoring the Health of an Access Gateway," on page 553.
`curl`	Use it to view identity provider metadata from the Linux Access Gateway. See Section 38.2.6, "Test Whether the Provider Can Access the Metadata," on page 594.
`tail -f`	Use it to view real time activity in key log files. For information on useful files to tail, see "Useful Troubleshooting Files" on page 639.
proc	Use it to check resources available on the system.
`netstat /ss`	Use it to view statistics about the listeners on the Linux Access Gateway.
`netcat`	Use it to access the Linux Access Gateway console, which displays statistics and information about various processes. For more information, see "The Linux Access Gateway Console" on page 637.
`tcpdump`	Use it to capture data on standard and loopback interfaces and to view SSL data with imported keys.
`nash`	Use it to manually configure log level verbosity and replace IP addresses. For log level information, see "Linux Access Gateway Logs" on page 669.
`/etc/init.d/novell-vmc`	Use the `novell-vmc` command line options to restart the proxy and view status. For more information, see Table 40-2 on page 637.
The `/chroot/lag/opt/novell/bin` directory contains the following scripts:	

Tool	Description
`getlaglogs.sh`	Generates a `/var/log/laglogs.tar.gz` file of the install and system log files. For more information, see "Linux Access Gateway Logs" on page 642.
`lagupgrade.sh`	Use it to apply patches. For more information, see "Upgrading the Linux Access Gateway" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.
`lagconfigure.sh`	Use it to resolve auto-import issues. For more information, see "Triggering an Import Retry" in the *Novell Access Manager 3.0 SP3 IR2 Installation Guide*.

You can use the following commands to stop and start the Linux Access Gateway and to view its status.

Table 40-2 *novell-vcm Commands*

Command	Description
`/etc/init.d/novell-vmc start`	Starts the Linux Access Gateway.
`/etc/init.d/novell-vmc stop`	Stops the Linux Access Gateway.
`/etc/init.d/novell-vmc status`	Displays the Linux Access Gateway status.
`/etc/init.d/novell-vmc restart`	Stops and starts the Linux Access Gateway.

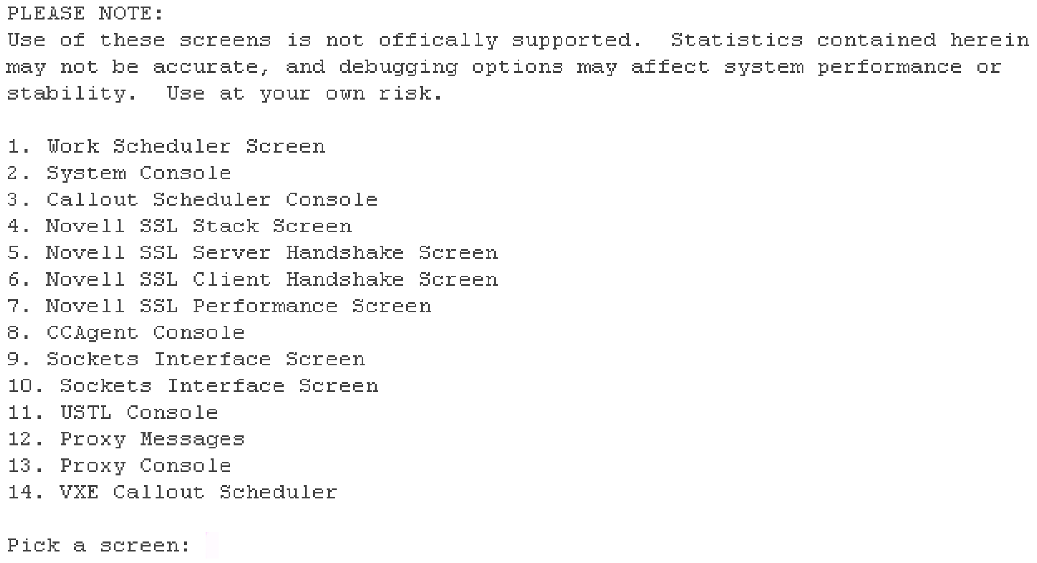
The Linux Access Gateway Console

1. To access the console, run the following command:

   ```
   netcat localhost 2300
   ```

2. Press Enter at the `Please enter terminal type` prompt.

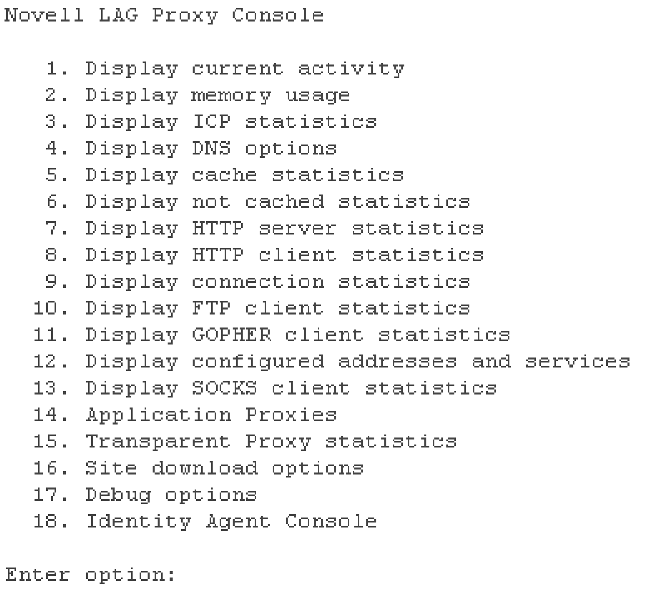
   This displays the Linux Access Gateway console screens.



```
PLEASE NOTE:
Use of these screens is not offically supported.  Statistics contained herein
may not be accurate, and debugging options may affect system performance or
stability.  Use at your own risk.

1. Work Scheduler Screen
2. System Console
3. Callout Scheduler Console
4. Novell SSL Stack Screen
5. Novell SSL Server Handshake Screen
6. Novell SSL Client Handshake Screen
7. Novell SSL Performance Screen
8. CCAgent Console
9. Sockets Interface Screen
10. Sockets Interface Screen
11. USTL Console
12. Proxy Messages
13. Proxy Console
14. VXE Callout Scheduler

Pick a screen:
```

Most of the time, the Proxy Console screen is the one you should pick. The other screens are used mainly by the developers of the Linux Access Gateway. If you are having SSL connection problems, the SSL screens can help in diagnosing the problem.

3 To access the Proxy Console screen, enter 13.



```
Novell LAG Proxy Console

   1. Display current activity
   2. Display memory usage
   3. Display ICP statistics
   4. Display DNS options
   5. Display cache statistics
   6. Display not cached statistics
   7. Display HTTP server statistics
   8. Display HTTP client statistics
   9. Display connection statistics
  10. Display FTP client statistics
  11. Display GOPHER client statistics
  12. Display configured addresses and services
  13. Display SOCKS client statistics
  14. Application Proxies
  15. Transparent Proxy statistics
  16. Site download options
  17. Debug options
  18. Identity Agent Console

Enter option:
```

4 To access a specific screen, enter the number.

Screen	Description
1. Display current activity	Displays information about connections (server and client), cached objects, and HTTP requests.

Screen	Description
2. Display memory usage	Displays information about memory pools and memory used and the types of objects stored in memory.
3. Display ICP statistics	Displays statistics for the Internet Cache Protocol.
4. Display DNS options	Displays statistics and information about the entries in the DNS table.
5. Display cache statistics	Displays information about cached objects and the COS partition. For more information, see "Checking if the COS Partition Is Mounted" on page 652.
6. Display not cached statistics	Displays statistics about requests for objects that cannot be cached.
7. Display HTTP server statistics	Displays statistics about the server handling of HTTP requests.
8. Display HTTP client statistics	Displays statistics about the client handling of HTTP requests.
9. Display connection statistics	Displays general information about connections.
10. Display FTP client statistics	Displays statistics about FTP client requests.
11. Display GOPHER client statistics	Displays statistics about GOPHER requests.
12. Display configured addresses and services	Displays information about the IP addresses that the Access Gateway is using.
13. Display SOCKS client statistics	Displays statistics about SOCKS client requests.
14. Application Proxies	Displays proxy service statistics.
15. Transparent Proxy statistics	Displays transparent proxy statistics.
16. Site download options	Displays information about the last download and prompts for information to schedule a new download.
17. Debug options	Allows you to control cache purging.
18. Identity Agent Console	Displays user information. For more information about the user screen, see "User Details" on page 647.

5. To return to the opening page of the console from other console page, press Esc+Enter.

   This key stroke works only on some pages.
6. To exit the console, press Ctrl+C.

Useful Troubleshooting Files

- "Viewing Log Files" on page 640
- "Using Touch Files" on page 641

Viewing Log Files

Table 40-3 describes the Linux Access Gateway files that contain troubleshooting information.

***Table 40-3** Log Files with Troubleshooting Information*

Log File	Description
`catalina.out`	Located in the `/var/opt/novell/tomcat4/logs` directory and available from the General Logging page in the Administration Console. The embedded service provide, which communicates with the Identity Server, writes to this log file. The log level is controlled by the Identity Server Configuration. For configuration information, see Section 39.1, "Turning on Logging for Policy Evaluation," on page 599. For information on how to use the entries for policy troubleshooting, see Chapter 39, "Troubleshooting Access Manager Policies," on page 599.
`ics_dyn.log`	Located in the `/var/log` directory and available from the General Logging page in the Administration Console. The proxy service writes to this log file. For information on enabling logging to this file, see "Linux Access Gateway Logs" on page 669. For maximum verbosity, the proxy service must be started in debug mode. See Table 40-2, "novell-vcm Commands," on page 637.
`lagsoapmessages`	Located in the `/var/log` directory and available from the General Logging page in the Administration Console. When enabled, this file contains a log of the SOAP messages between the Linux Access Gateway and the embedded service provider for authentication (roles, contracts, and timeouts) and policy interaction (Authorization, Form Fill, and Identity Injection). For information on enabling logging to this file, see "Configuring Logging of SOAP Messages and HTTP Headers" on page 672.
`laghattpheaders`	Located in the `/var/log` directory and available from the General Logging page in the Administration Console. When enabled, this file contains a log of the HTTP headers to and from the Linux Access Gateway. For information on enabling logging to this file, see "Configuring Logging of SOAP Messages and HTTP Headers" on page 672.

Using Touch Files

Table 40-4 describes the touch files that control how the Linux Access Gateway starts.

Table 40-4 *Touch Files*

Filename	Description
`.~newInstall`	Located in the `/var/novell` directory. Purges cache during restart. The Linux Access Gateway creates this file during every start. If you want the Linux Access Gateway to come up without the contents cached in the previous run, remove this file before you restart the Linux Access Gateway.
`.~fastAutoRestart`	Located in the `/var/novell` directory. Restarts the Linux Access Gateway in the daemon mode without the VMController. This automatically removes the /var/novell/.~newInstall file during the Linux Access Gateway restart. This ensures that the contents cached in the previous run are not read and the time taken to restart the Linux Access Gateway is reduced.

`.modVia`	Located in the `/var/novell` directory.
	Adds the device ID in the Via header that is sent by the Linux Access Gateway to the Web server.
	To add a device ID in the Via header, enter the following command:
	`/var/novell/.modVia`
	Then restart the Linux Access Gateway.
	The Linux Access Gateway sends the Via header in the following format:
	`Via: 1.0 www.mylag.com (Access Gateway 3.0.1-72-D06FBFA8CF21AF45)`

`.ForceHTTPSSchemeInESPRedirection`	Located in the `/var/novell` directory. Forces the Linux Access Gateway to always return the URL in the HTTPS schema.
`.EnableSecureCookie`	Located in the `var/novell/` directory. It sets the HTTPS services authentication cookie with the keyword `secure`.
`.ForceSecureCookie`	Located in the `/var/novell/` directory. When this touch file is present, even the HTTP services have the authentication cookie set with the keyword `secure`.
`.EnableHttpOnlyCookie`	Located in the `/var/novell` directory. Configures the Linux Access Gateway to set its authentication cookie with the `HttpOnly` keyword, to prevent scripts from accessing the cookie.

40.2.2 Troubleshooting a Failed Linux Access Gateway Configuration

If the IP address and other network configurations are not reflected in the installed Linux Access Gateway, log in as a `root` user and run the following commands:

```
rm /opt/novell/legacy/etc/proxy/.novell_lag_lock
/etc/init.d/novell-vmc stop
/etc/init.d/novell-vmc start
```

40.2.3 Troubleshooting a Linux Access Gateway Crash

The Linux Access Gateway might have crashed because of the following reasons:

- SIGSEGV
- ASSERT (for a debug build only)

The following sections explain how to gather the files that need to be sent to Novell for a resolution of the problem.

- "Linux Access Gateway Logs" on page 642
- "Event Log" on page 642
- "Core Dump" on page 644
- "Proxy Hang Core" on page 645
- "Packet Capture" on page 645

Linux Access Gateway Logs

1. Enter the following command from the bash shell to collect the debug log files that are generated:

   ```
   /chroot/lag/opt/novell/bin/getlaglogs.sh
   ```

2. The `laglogs.tgz` tar file is located in the `/var/log` directory.
3. Send this tar file to Novell Support.

Event Log

By default the event log size is 15 MB. The size of event log can be controlled by configuring the required event log size in the `eventlogsize.cfg` file, located at the `/chroot/lag/etc/opt/novell` directory. For example, if you specify 350 in the file, you can configure an event log of size 350 MB. This file should contain only the file size information. This file should not contain any other characters or new lines.

The procedure for obtaining the event log depends upon the build type:

- "Event Log for a Production Build" on page 643
- "Event Log for a Debug Build" on page 643

Event Log for a Production Build

To get the event log for the production build:

1 Log in as the `root` user.

2 To disconnect all instances of Linux Access Gateway, enter the following command:

```
/etc/init.d/novell-vmc stop
```

3 Enter the following command to change the root environment:

```
chroot /chroot/lag
```

4 To start the process, enter the following command:

```
gdb /opt/novell/bin/ics_dyn 2>/var/log/ics_dyn.log
```

5 At the GDB prompt, run the following command:

```
run -m <memory>
```

Where <*memory*> is the percentage of total memory to be used for ics_dyn process. It is recommended to set this value in the range of 20-30 percent.

6 Repeat the scenarios to reproduce the issue.

  6a If you are trying to reproduce the proxy crash, you see the GDB prompt as soon as the crash is reproduced.

  6b If you are trying to reproduce a functionality issue, press Crtl+C to enter the GDB prompt as soon as the issue is reproduced.

  For a list of commands that can be entered in the debugger, see "Useful Debugger Commands" on page 644.

7 To save event logs to a file, enter the following command:

```
d ,save 1
```

This stores all the events in the `/chroot/lag/opt/novell/debug/<pid>all_events.0.txt` file.

8 Tar or Zip this file and send it to Novell Support.

Event Log for a Debug Build

To get the event log:

1 Log in as the `root` user.

2 To stop all instances of Linux Access Gateway, enter the following command:

```
/etc/init.d/novell-vmc stop
```

3 To start the Novell Linux Access Gateway in debugging mode, enter the following command:

```
/etc/init.d/novell-vmc gdb
```

4 To run the Linux Access Gateway process, enter the following command at the GDB prompt:

```
run -m <memory> 2>/var/log/ics_dyn.log
```

Where <*memory*> is the percentage of total memory to be used for ics_dyn process. It is recommended to set this value in the range of 20-30 per cent.

5 Repeat the scenarios to reproduce the issue.

  5a If you are trying to reproduce the proxy crash, you will enter the GDB prompt as soon as the crash is reproduced.

5b If you are trying to reproduce a functionality issue, enter the following command to enter the GDB prompt as soon as the issue is reproduced:

```
Crtl+C
```

NOTE: For a list of commands that can be entered in the debugger, see "Useful Debugger Commands" on page 644.

---

6. To save all event logs to a file, enter the following command:

   ```
   d ,save 1
   ```

   This stores all the events in the `/chroot/lag-debug/opt/novell/debug/<pid>all_events.0.txt` file.

7. Tar or Zip this file and send it to Novell Support.

Useful Debugger Commands

***Table 40-5** GDB Commands*

Command	Function
`gcore`	Generate core file
`k`	Kill process
`q`	Quit GDB prompt
bt	Print the back trace

Core Dump

Before you begin, make sure there is free space in `root` to hold the core file and that the space is at least equal to the RAM size

To collect a core dump:

1. Log in as the `root` user.
2. To disconnect all instances of the Linux Access Gateway, enter the following command:

   ```
   /etc/init.d/novell-vmc stop
   ```

3. At the bash prompt, specify the following command:

   ```
   touch /tmp/.dumpcore
   ```

4. Enter the following command to start the Linux Access Gateway:

   ```
   /etc/init.d/novell-vmc start
   ```

5. Repeat the scenarios to reproduce the issue.

   The core is dumped to the `/chroot/lag` `core.<pid>` file.

   `<pid>` is the process ID of ics_dyn process.

   After the core is dumped, the Linux Access Gateway restarts.

6. Tar or Zip the core dump send it to Novell Support.

Proxy Hang Core

To analyze the proxy hang and create a core file:

1. Enter the following command to change the root environment:
   ```
   chroot /chroot/lag
   ```
2. Enter the following command to attach the ics_dyn process to the debugger:
   ```
   gdb /opt/novell/bin/ics_dyn <pid>
   ```
   Where <*pid*> refers to the Process ID of the ics_dyn process.
3. At the GDB prompt, enter the following command:
   ```
   set logging on <filename>
   ```
   Where <*filename*> specifies the name of the file that will store the output of the executed debugger commands.
4. Enter the following command to collect a stack trace of all threads:
   ```
   thread apply all bt
   ```
5. Enter the following command to turn off logging:
   ```
   set logging off
   ```
6. Enter the following command to save the core dump in the `/chroot/lag` directory.
   ```
   gcore
   ```
   The core dump is saved as `core.<pid>`
7. Tar or Zip this file and send it to Novell Support.

Packet Capture

The `tcpdump` utility allows you to capture network trace packets.

1. Log in as the `root` user.
2. Enter the following command:
   ```
   tcpdump -s0 -n -t -p -i ‘any’ -w filename.cap
   ```
3. Tar or Zip this file and send it to Novell Support.

40.2.4 HTTP Requests Are Dropped

If you have HTTP requests that are being dropped, create the following touch file: `/var/novell/.AllowUnknownHTTPMethods`

This file allows the Access Gateway to forward unknown HTTP methods.

Restart the Access Gateway.

If this solves your problem, discover the name of the unknown HTTP method and inform Novell so that it can be added to the list of supported methods. To allow any unknown method to be forwarded, opens a security vulnerability.

40.2.5 Linux Access Gateway Not Responding

If the Linux Access Gateway is not responding, do the following:

1. Enter the following command to change the root environment:

   ```
   chroot /chroot/lag
   ```

2. Enter the following command to attach the ics_dyn process to the debugger:

   ```
   gdb /opt/novell/bin/ics_dyn <pid>
   ```

   Where <*pid*> refers to the Process ID of the ics_dyn process. You can get the process ID by entering the following command:

   ```
   pgrep ics_dyn
   ```

3. At the GDB prompt, enter the following command:

   ```
   set logging file <filename>
   ```

   Where <*filename*> specifies the name of the file that will store the output of the executed debugger commands.

4. Enter the following command to start logging:

   ```
   set logging on
   ```

5. Enter the following command to collect a stack trace of all threads:

   ```
   thread apply all bt full
   ```

6. Enter the following command to turn off logging:

   ```
   set logging off
   ```

7. Enter the following command to save the core dump in the `/chroot/lag` directory.

   ```
   gcore
   ```

   The core dump is saved as `core.<pid>`

8. Tar or Zip this file and send it to Novell Support.

40.2.6 Connection Details

To obtain connection information:

1. Log in as the `root` user.
2. At the bash prompt, enter one of the following `netstat` commands:

Command	**Details**
`netstat -anp`	Provides the connection information
`netstat -s -t`	Provides the connection statistics

40.2.7 Network Socket Issues

This section lists various issues related to network sockets and provides information on how to verify bind and connection issues:

- "Socket Listener Bind" on page 647

- “Issues with Outgoing Connections” on page 647

Socket Listener Bind

To verify whether the socket listener is bound to the required port:

1. Log in as the `root` user.
2. At the bash prompt, enter the following command:
   ```
   netstat -anp | grep LISTEN
   ```
   All ports are displayed.
3. Search for the desired port.

   If the required port is not visible in the list, a bind failure has occurred.

Issues with Outgoing Connections

To verify that the Access Gateway is able to make outbound connections:

1. Log in as the `root` user.
2. At the bash prompt, view the following log file:
   ```
   /var/log/ics_dyn.log
   ```
3. Search for a connection message. If the service is unavailable, the file contains messages similar to the following:
   ```
   ERROR Connection FAILED with peer
   ```

40.2.8 Authentication Issues

This section provides information related to authentication:

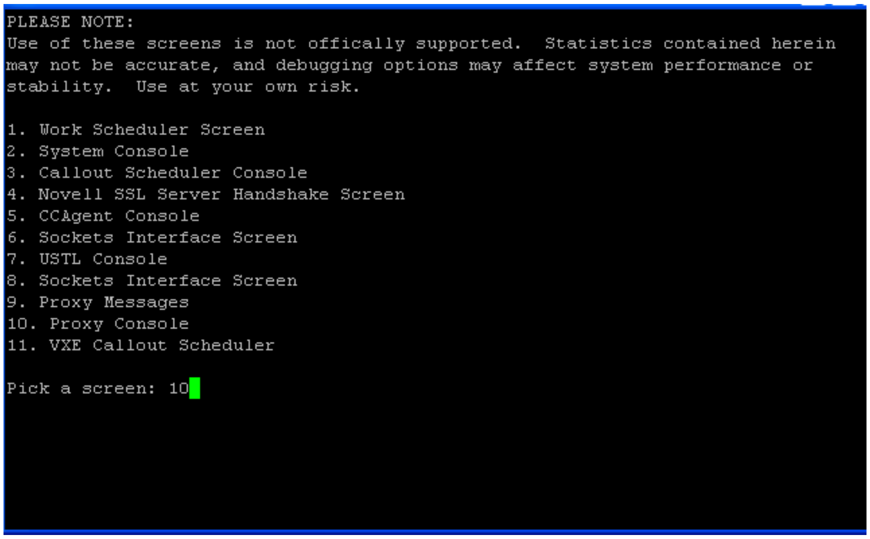
- “User Details” on page 647
- “Error Codes” on page 649

User Details

To check the details about the users logged in to the Linux Access Gateway:

1. To access the console, enter the following command:
   ```
   netcat localhost 2300
   ```
2. Press Enter at the `Please enter terminal type` prompt.

   This displays the Linux Access Gateway console screens.

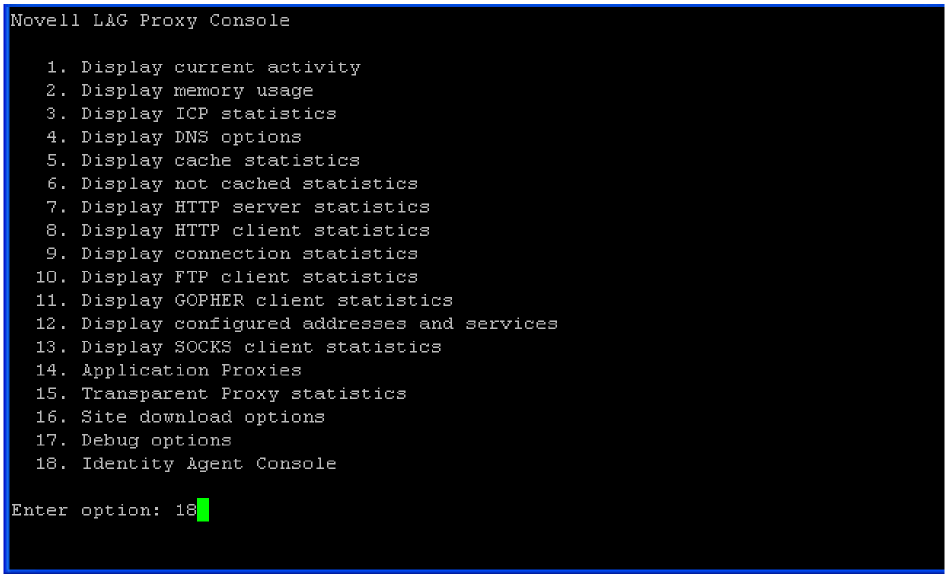


```
PLEASE NOTE:
Use of these screens is not offically supported.  Statistics contained herein
may not be accurate, and debugging options may affect system performance or
stability.  Use at your own risk.

1. Work Scheduler Screen
2. System Console
3. Callout Scheduler Console
4. Novell SSL Server Handshake Screen
5. CCAgent Console
6. Sockets Interface Screen
7. USTL Console
8. Sockets Interface Screen
9. Proxy Messages
10. Proxy Console
11. VXE Callout Scheduler

Pick a screen: 10
```

3 Enter the *Proxy Console* option number at the *Pick a Screen* prompt.

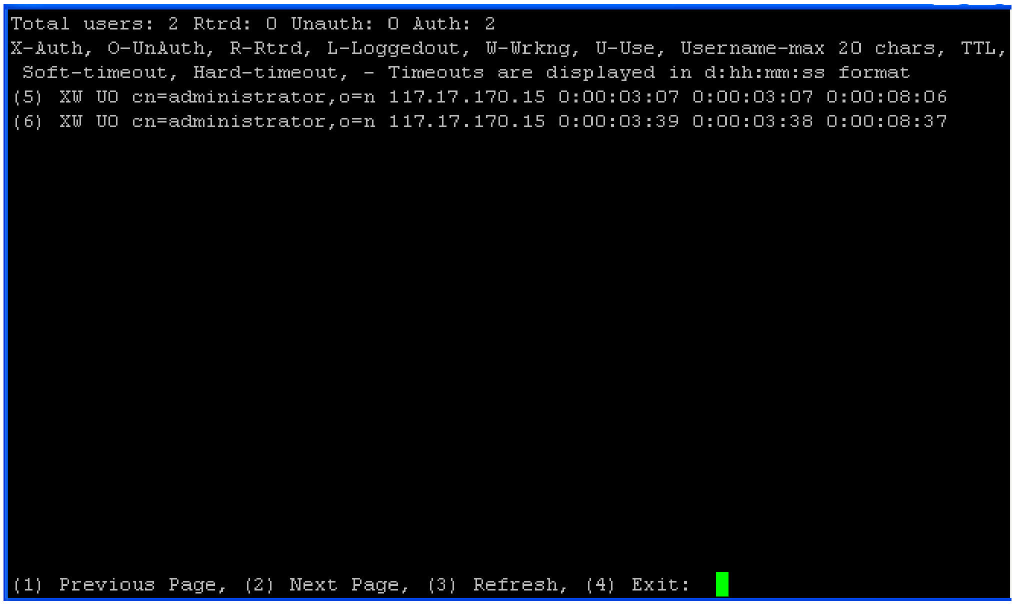
The Linux Access Gateway Console screen is displayed.

4 To select the *Identity Agent Console* option, enter the option number at *Enter Option*.



```
Novell LAG Proxy Console

   1. Display current activity
   2. Display memory usage
   3. Display ICP statistics
   4. Display DNS options
   5. Display cache statistics
   6. Display not cached statistics
   7. Display HTTP server statistics
   8. Display HTTP client statistics
   9. Display connection statistics
  10. Display FTP client statistics
  11. Display GOPHER client statistics
  12. Display configured addresses and services
  13. Display SOCKS client statistics
  14. Application Proxies
  15. Transparent Proxy statistics
  16. Site download options
  17. Debug options
  18. Identity Agent Console

Enter option: 18
```

The Identity Agent Console screen is displayed.



```
Total users: 2 Rtrd: 0 Unauth: 0 Auth: 2
X-Auth, O-UnAuth, R-Rtrd, L-Loggedout, W-Wrkng, U-Use, Username-max 20 chars, TTL,
 Soft-timeout, Hard-timeout, - Timeouts are displayed in d:hh:mm:ss format
(5)  XW UO cn=administrator,o=n 117.17.170.15 0:00:03:07 0:00:03:07 0:00:08:06
(6)  XW UO cn=administrator,o=n 117.17.170.15 0:00:03:39 0:00:03:38 0:00:08:37









(1) Previous Page, (2) Next Page, (3) Refresh, (4) Exit:
```

The user information contains the following items:

- **X:** An authenticated user.
- **O:** An unauthenticated user.
- **R:** A retired user; the user session has timed out. The default time-out is 3 minutes. In this state, the user session is deleted. If the user makes another request from the browser session, the Linux Access Gateway requires the user to authenticate.
- **L:** The user has logged out of the session.
- **W:** The user session is functional.
- **U:** The use count is more than zero.
- **Username:** The full distinguished name of the user. The username can contain a maximum of 20 characters.
- **TTL:** The time remaining before the user session goes to the retired state if the user session remains idle.
- **Timeout:** The session timeout is displayed in d:hh:mm:ss format.

The screen displays 20 users at a time. The screen also displays the browser IP address. The following options are available at the bottom of the screen:

- **Previous Page:** Lets you go to the previous page.
- **Next Page:** Lets you go to the next page (to view the next set of users).
- **Refresh:** Refreshes the page to reflect the latest user status.
- **Exit:** Exits the console.

Error Codes

The following error codes indicate authentication problems:

500 Internal Server Error

Possible Cause: Authentication failed because of a system error.

Action: Contact Novell Support.

504 Gateway Timed Out

Possible Cause: The authentication back-end channel is not working.

Action: Check to see if the embedded service provider is listening on the loopback address 127.0.0.1 at port 8080: Use the following command:

```
netstat -na | grep 8080
```

If the embedded service provider is down, restart the service provider from the Administration Console.

If the issue persists, contact Novell Support.

40.2.9 Rewriter Issues

The following sections explain how to troubleshoot problems with the rewriter:

- "Reading Configuration Files" on page 650
- "Rewriting a URL" on page 650

Reading Configuration Files

If the rewriter is successful in reading the configuration files, and you have enabled the log level to LOG_INFO, the following message is displayed in the /var/log/ics_dyn.log file:

```
Reading Config File
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:Configuration information
read successfully
```

For more information on configuring log levels, see "Configuring Log Levels" on page 670.

If the rewriter fails to read the configuration files, the following message is displayed:

```
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:Reading configuration
failed for ssTypeName=www.mynovell.com
```

If this happens, re-create the corresponding proxy service and restart the Linux Access Gateway service.

Rewriting a URL

Set the log level to LOG_DEBUG to view rewriter log messages in the `/var/log/ics_dyn.log` file. (See "Configuring Log Levels" on page 670.)

For example, if the Rewriter successfully rewrites the URL, the following messages are displayed:

```
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/common/inc/nav/main.js' Content type match, Will
Rewrite
```

```
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/common/inc/nav/main.js' Unknown Content-Type -
automatic match - Will Rewrite
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0::'http://
www.mynovell.com:9090/common/inc/nav/main.js' NULL Content-Type -
automatic match - Will Rewrite
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:In
RewriterOption::shouldRewriteUrl, returning TRUE.
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/common/inc/nav/main.js' Unknown extension -
automatic match - Will Rewrite
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/common/inc/nav/main.js' NULL extension -
automatic match - Will Rewrite
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/common/inc/nav/main.js' Extension type match -
Will Rewrite
```

If the conditions for rewriting a URL fail, the following messages are displayed:

```
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:URL:'http://
www.mynovell.com:9090/favicon.ico' - Did not match INCLUDE list,
Content-Type and Extension type
Aug 16 04:16:51 proxy140 LINUX_AG:REWRITER:0:In
RewriterOption::shouldRewriteUrl, returning FALSE.
```

Check the rewriter configuration. Ensure that your content type, extension type, and include URL list are valid.

40.2.10 COS Related Issues

The following sections explain how to troubleshoot COS (cache object store) partition issues:

- "Viewing COS Partition Details" on page 651
- "Checking if the COS Partition Is Mounted" on page 652

Viewing COS Partition Details

You can view COS partition details either through YaST or through the nash prompt.

Using YaST

1. Log in as the `root` user.
2. At command prompt, enter the following command:

   ```
   fdisk -l
   ```

   The partition details are displayed. Check for COS partition details. Make sure that a partition is created with a partition ID of `68` and that the file system is created as type `unknown`.

Using nash

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

```
nash
```

2. At the `nash` shell prompt, enter the following command:

   ```
   configure .current
   ```

3. Enter the following command:

   ```
   vm scan
   ```

   If the COS partition is already created, the details are displayed.

Checking if the COS Partition Is Mounted

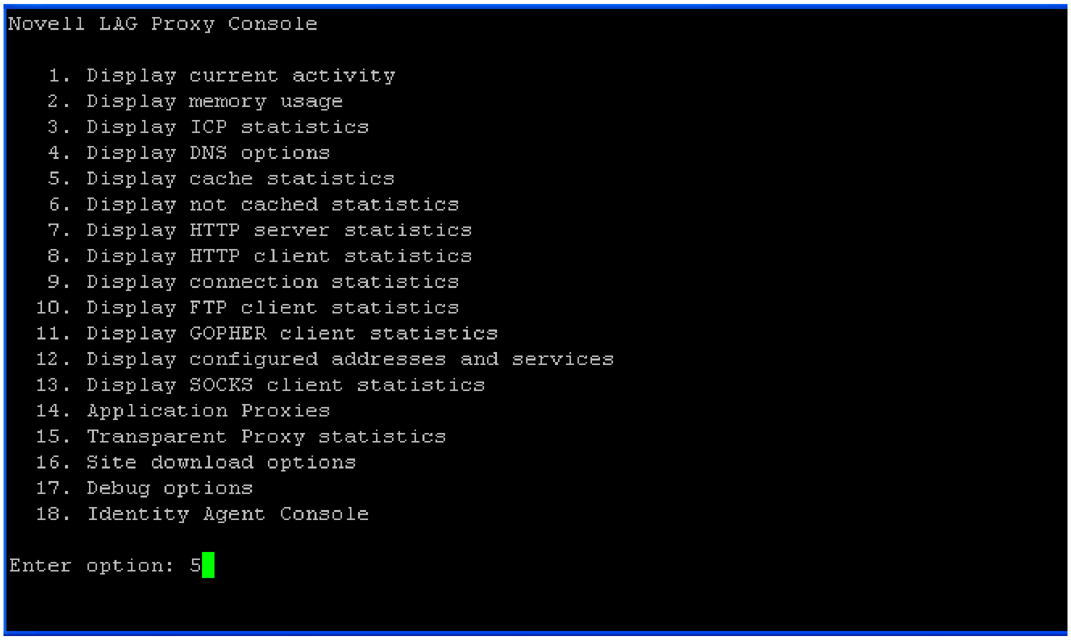
1. Access the Linux Access Gateway main screen.

   For more information on how to access the Linux Access Gateway main screen, see Section , "The Linux Access Gateway Console," on page 637.

2. Enter the *Proxy Console* option number at the *Pick a Screen* prompt.

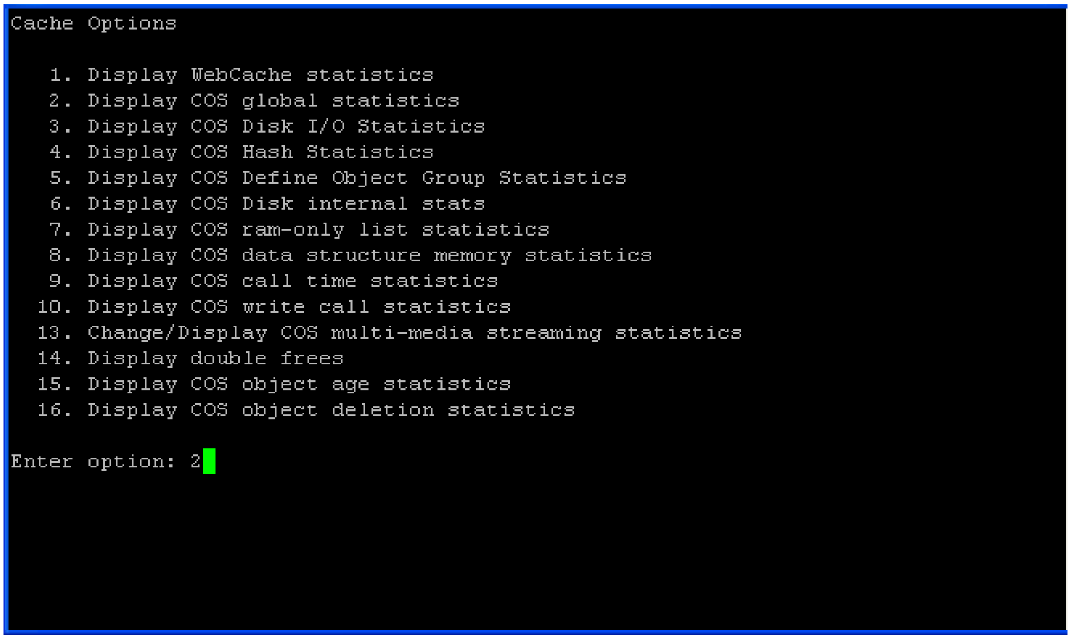
   The Linux Access Gateway Console screen is displayed.

3. Enter the *Display Cache Statistics* option number at the *Enter option* prompt.



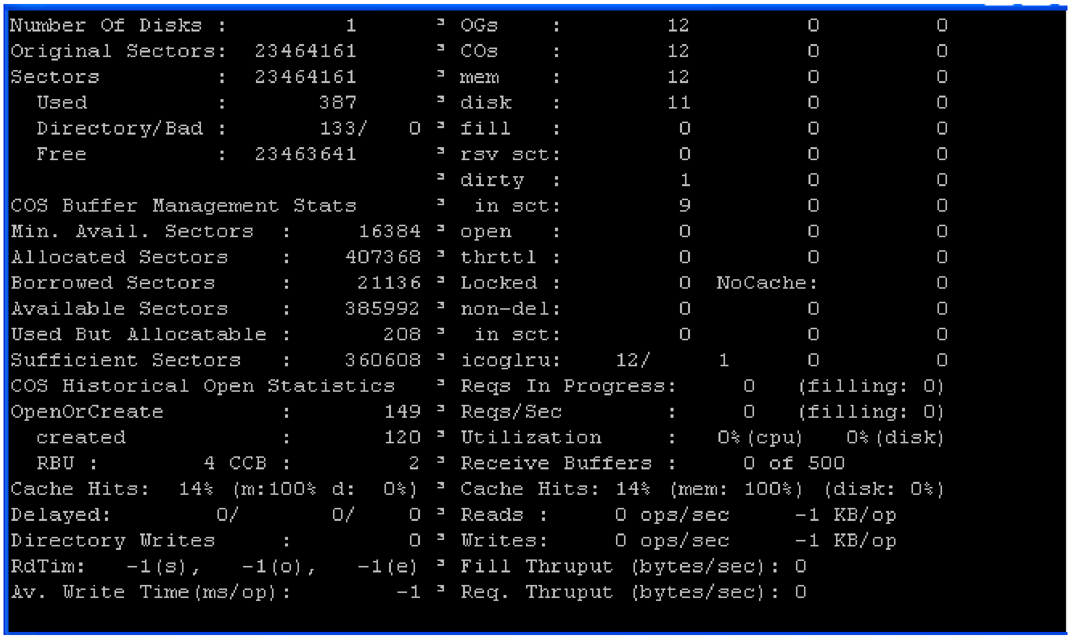
```
Novell LAG Proxy Console

   1. Display current activity
   2. Display memory usage
   3. Display ICP statistics
   4. Display DNS options
   5. Display cache statistics
   6. Display not cached statistics
   7. Display HTTP server statistics
   8. Display HTTP client statistics
   9. Display connection statistics
  10. Display FTP client statistics
  11. Display GOPHER client statistics
  12. Display configured addresses and services
  13. Display SOCKS client statistics
  14. Application Proxies
  15. Transparent Proxy statistics
  16. Site download options
  17. Debug options
  18. Identity Agent Console

Enter option: 5
```

4. Enter the *Display COS Global Statistics* option number at the *Enter option* prompt.



```
Cache Options

   1. Display WebCache statistics
   2. Display COS global statistics
   3. Display COS Disk I/O Statistics
   4. Display COS Hash Statistics
   5. Display COS Define Object Group Statistics
   6. Display COS Disk internal stats
   7. Display COS ram-only list statistics
   8. Display COS data structure memory statistics
   9. Display COS call time statistics
  10. Display COS write call statistics
  13. Change/Display COS multi-media streaming statistics
  14. Display double frees
  15. Display COS object age statistics
  16. Display COS object deletion statistics

Enter option: 2
```

The following details are displayed if the COS partition is mounted:



```
Number Of Disks :         1      ³ OGs    :        12         0         0
Original Sectors:  23464161      ³ COs    :        12         0         0
Sectors         :  23464161      ³ mem    :        12         0         0
  Used          :       387      ³ disk   :        11         0         0
  Directory/Bad :       133/   0 ³ fill   :         0         0         0
  Free          :  23463641      ³ rsv sct:         0         0         0
                                 ³ dirty  :         1         0         0
COS Buffer Management Stats      ³  in sct:         9         0         0
Min. Avail. Sectors  :     16384 ³ open   :         0         0         0
Allocated Sectors    :    407368 ³ thrttl :         0         0         0
Borrowed Sectors     :     21136 ³ Locked :         0  NoCache:         0
Available Sectors    :    385992 ³ non-del:         0         0         0
Used But Allocatable :       208 ³  in sct:         0         0         0
Sufficient Sectors   :    360608 ³ icoglru:    12/     1      0         0
COS Historical Open Statistics   ³ Reqs In Progress:     0    (filling: 0)
OpenOrCreate         :       149 ³ Reqs/Sec        :     0    (filling: 0)
  created            :       120 ³ Utilization     :   0%(cpu)   0%(disk)
  RBU :         4 CCB :        2 ³ Receive Buffers :     0 of 500
Cache Hits:  14% (m:100% d:   0%) ³ Cache Hits: 14% (mem: 100%) (disk: 0%)
Delayed:         0/        0/    0 ³ Reads :     0 ops/sec      -1 KB/op
Directory Writes     :         0 ³ Writes:     0 ops/sec      -1 KB/op
RdTim:   -1(s),   -1(o),   -1(e) ³ Fill Thruput (bytes/sec): 0
Av. Write Time(ms/op):        -1 ³ Req. Thruput (bytes/sec): 0
```

40.2.11 Memory Issues

The following sections explain how to troubleshoot memory issues:

- "Checking Memory Details and Related Information" on page 654
- "Checking Available Memory" on page 654

Checking Memory Details and Related Information

Most of the information, including the memory details, can be accessed by entering the following command at the `bash` prompt:

```
top
```

Ensure that the Linux Access Gateway does not occupy more than the percentage of the memory requirements you set. ics_dyn occupies approximately 20 to 25 percent of the total memory by default.

Levels	Requirement
Lower Limit	5 Percent
Requirement for Access Gateway	500 MB
Upper Limit	80 percent
Default	20 percent

Checking Available Memory

As the `root` user, enter the following command at the bash prompt:

```
cat /proc/meminfo | grep MemTotal
```

40.2.12 Authorization and Identity Injection Issues

- "Authorization and Identity Injection Error Messages" on page 654
- "Identity Injection Failures" on page 655

Authorization and Identity Injection Error Messages

If you have already configured the Identity Injection policies, you might receive the following errors while trying to send a browser request:

- `Service provider is in halted state. Please contact your administrator to restart Service Provider from Administrator Console.`
- `Policy engine is sending invalid response. Please contact your administrator to restart Service Provider from Administrator Console.`
- `Unable to process your request.`
- `Unable to process your request due to parseXML failure.`

These errors indicate that the embedded service provider is down. Every Identity Injection policy has a policy ID, which is sent to the Access Gateway by the embedded service provider. If the embedded service provider is down, the Access Gateway does not get the policy ID, and an error is thrown. Restart the embedded service provider from the Administration Console as follows:

1. In the Administration Console, click *Access Manager* > *Access Gateways*.
2. Select the server, then click *Actions*.

3 Click *Service Provider* > *Restart Service Provider*.

4 Click *OK*.

Identity Injection Failures

Identity injection might fail while trying to inject authentication headers because of improper policy configuration or because the Identity Server is not sending values to the Access Gateway.

Check the `/var/log/ics_dyn.log` file for the following error messages:

- `Customer Header Injection Failed.`
- `Query String Injection Failed.`
- `Authentication Header Injection Failed.`

To receive help resolving identity injection failures, send the following information to Novell Support:

- Linux Access Gateway logs. For more information on how to get Linux Access Gateway log files, see "Linux Access Gateway Logs" on page 669.
- Packet Capture. For more information on how to get packet captures, see "Packet Capture" on page 645.

40.2.13 Form Fill Issues

Form Fill error messages are logged only if you set the log level to LOG_DEBUG. The entries are logged in the `ics_dyn.log` file. Search for entries with a correlation tag of `AM#504507`. For more information, see Section 39.2.6, "Form Fill Traces," on page 614.

This section contains the following information about form fill issues:

- "Alert: SSO (Form Fill) Failed Due to Malformed HTML" on page 655
- "Form Fill Error Messages" on page 656
- "Form Fill Failure Because of Incorrect Policy Configuration" on page 656
- "Browser Spinning Issues" on page 656

Alert: SSO (Form Fill) Failed Due to Malformed HTML

Sometimes you might get the following error message:

```
Alert: SSO (Form Fill) Failed Due to Malformed HTML
```

The cause and action for that error could be the following:

Possible Cause: If this message appears on the login page which was to be filled by the Linux Access Gateway Form Fill, then the HTML page is malformed.

Action: You have to manually fill the form.

Possible Cause: If this message is displayed in any page other than the login page that was to be filled by the Linux Access Gateway, then this implies that the CGI or the page matching criteria configured for the Linux Access Gateway Form Fill policy matched the other pages and that there was a failed attempt to fill those pages.

Action: Check and modify the CGI and the Page Matching Criteria in the policy in such a way that the policy is applied only to the login page that you want the Linux Access Gateway to fill.

Form Fill Error Messages

You might get the following errors when sending a browser request:

- `DataStore Error`
- `The service provider is not running at the moment. Please retry after a few seconds.`

These errors indicate that the Access Gateway cannot retrieve the information that is essential to process the browser request, or is unable to save the information provided by the user because the embedded service provider is down. Retry the action after a few seconds. If the error persists, restart the embedded service provider from the Administration Console.

Form Fill Failure Because of Incorrect Policy Configuration

Form fill fails if the policy is not configured correctly. For configuration information, see Chapter 30, "Creating Form Fill Policies," on page 481.

Browser Spinning Issues

Browser spinning can occur if inappropriate data is filled in the form because of one of the following reasons:

- Shared secrets are configured, the user provided incorrect data to the Linux Access Gateway, and there are no appropriate actions configured to handle login failure.
- A Credential Profile with LDAP attributes has been configured, and there is a mismatch between the user name used to authenticate to the Linux Access Gateway and the user name used to authenticate to the accelerated Web server.

When a Form Fill policy succeeds and the authentication to the Web server fails, the Web server redirects the browser to its authentication page again and again, if auto-submit is enabled. In such a situation, if there is no appropriate login-failure action configured in the policy, the browser "spins" endlessly.

If this happens, do the following:

- Kill the browser session. If you are unable to do this, run the following commands to restart the Linux Access Gateway:

  ```
  /etc/init.d/novell-vmc stop
  /etc/init.d/novell-vmc start
  ```
- If the issue is with a Credential Profile with LDAP attributes, verify which LDAP attributes are required by the Web server, and create the appropriate entries in the Form Fill policy.
- If the issue is with shared secrets, delete the corresponding values from the Secret Store. If it is not possible to delete the value, modify the corresponding policy to use a different or a new custom attribute or shared secret attribute. For more information on modifying the policy, see Section 30.3, "Implementing Form Fill Policies," on page 487.

40.3 Troubleshooting the NetWare Access Gateway

- Section 40.3.1, “Additional Options During the Boot Process,” on page 657
- Section 40.3.2, “Unlocking the NetWare Access Gateway Console,” on page 658
- Section 40.3.3, “Setting the Date and Time at the Console,” on page 658
- Section 40.3.4, “Command Line Options,” on page 659
- Section 40.3.5, “Telnet Fails after Performing an Upgrade,” on page 659
- Section 40.3.6, “SSL Certificate Error with X.509 Authentication from NetWare Access Gateway,” on page 659

40.3.1 Additional Options During the Boot Process

You can enter additional commands during the boot process to enable monitoring of the load process and local maintenance.

1. Boot the machine and wait for the following screen:

   ```
   =================================================================
   Loading Bootstrap ...
   Preparing to start NetWare ...

   Press any key to Interrupt
   =================================================================
   ```

2. Press any key. The following menu appears:

   ```
   =================================================================
   Default NetWare configuration file detected (CONFIG.NW) Contents:
   -LS 1024 -CON "Booting Novell(R) Access Gateway 3.0" -L

   Type:
   S to start NetWare
   P to specify additional starting parameters
   H for help
   Enter selection:
   ==================================================================
   ```

3. Enter P, then the following parameter:

   `-NetWareOnly`

4. To start NetWare, enter S.

   The NetWare Access Gateway boots to the NetWare prompt, so you can do local maintenance.

5. (Optional) During the blue screen where all the modules are counted in the load process, enter one of the following keystrokes:

   - To unlock this screen so you can see the loading process, press SHIFT+CTRL+ALT+U.
   - To boot to the NetWare command line, press SHIFT+CTRL+ALT+N.

   As a memory aid for the two key sequences, remember that U indicates unlock, and N indicates NetWare.

40.3.2 Unlocking the NetWare Access Gateway Console

Before you can enter NetWare commands or view the logger screen, you must unlock the console.

1. To unlock the console, enter

   ```
   unlock
   ```

2. When prompted for a password, press Enter.

   The console is now unlocked and the active screen is the device manager screen. From this screen you can enter device manager commands.

3. To switch to the logger screen or other NetWare screens, enter

   ```
   debug
   ```

4. When prompted for a password, enter

   ```
   proxydebug
   ```

5. To switch from the device manager screen, press Ctrl+Escape and enter the screen number.

40.3.3 Setting the Date and Time at the Console

If you inadvertently set the date and time on the Access Gateway to a time before the certificates are valid, the Administration Console is denied access to the Access Gateway and can no longer interact with it. To correct this problem, you must reset the date and time at the Access Gateway console.

1. Unlock the console.

   For instructions, see Section 40.3.2, "Unlocking the NetWare Access Gateway Console," on page 658.

2. Switch to the device manager screen.

3. Enter the following command:

   ```
   set date [year=<yyyy>,] [month=<mm>,] [day=<dd>,] [time=<hh:mm:ss>]
   ```

   Replace the variables with the following values:

<yyyy>	Replace with a four-digit value representing the current year, such as 2007.
<mm>	Replace with a two-digit value representing the current month, with 1 representing January and 12 representing December.
<dd>	Replace with a two-digit value, from 1 to 31, indicating the current day of the month.
<hh:mm:ss>	Replace hh with a two-digit value, from 1 to 24, indicating the current hour. Replace mm with a two-digit value, from 0 to 60, indicating the current minutes. Replace ss with a two-digit value, from 0 to 60, indicating the current seconds.

   For example, to set the date to December 1, 2006 and the time to 10:10 am, enter the following:

   ```
   set date year=2006, month=12, day=1, time=10:10:00
   ```

   The `set date` command disables NTP. Use the Administration Console to enable it.

40.3.4 Command Line Options

Access Manager has been designed to use the Administration Console for most management and configuration tasks. If you have created a group for your Access Gateways, Novell highly recommends that you use the Administration Console for these tasks.

The Access Gateway does not push configuration changes to the Administration Console. As soon as you make a change at the Administration Console and save the change, the Administration Console pushes the change to the Access Gateway and wipes any changes that have been made manually with the command line interface. Various troubleshooting tips explain how to use various command line options; other than troubleshooting, you should have very little cause to use them.

The NetWare Access Gateway uses the SET command syntax for its command line options. You must unlock the console to gain access to the command line prompt. (See Section 40.3.2, "Unlocking the NetWare Access Gateway Console," on page 658.)

To get a list of possible commands, enter the following command at the command line prompt:

```
help
```

To get help for a particular command, enter

```
help <command_name>
```

Replace <command_name> with the name of a command, such as `set`.

40.3.5 Telnet Fails after Performing an Upgrade

After preforming an over-the-wire upgrade, Telnet does not allow you to connect. To correct this problem, enter the following command at the NetWare Access Gateway console:

```
clear adminacl serveraddress
```

40.3.6 SSL Certificate Error with X.509 Authentication from NetWare Access Gateway

If you set up an X.509 contract and use it to authenticate from the NetWare Access Gateway, you might see an error generated for the SSL certificate, causing possible problems authenticating with certificates. This occurs during SSL re-negotiation between Tomcat and the Internet Explorer browser, and is possibly an Internet Explorer bug. This error does not occur when using Firefox.

Troubleshooting the SSL VPN

41

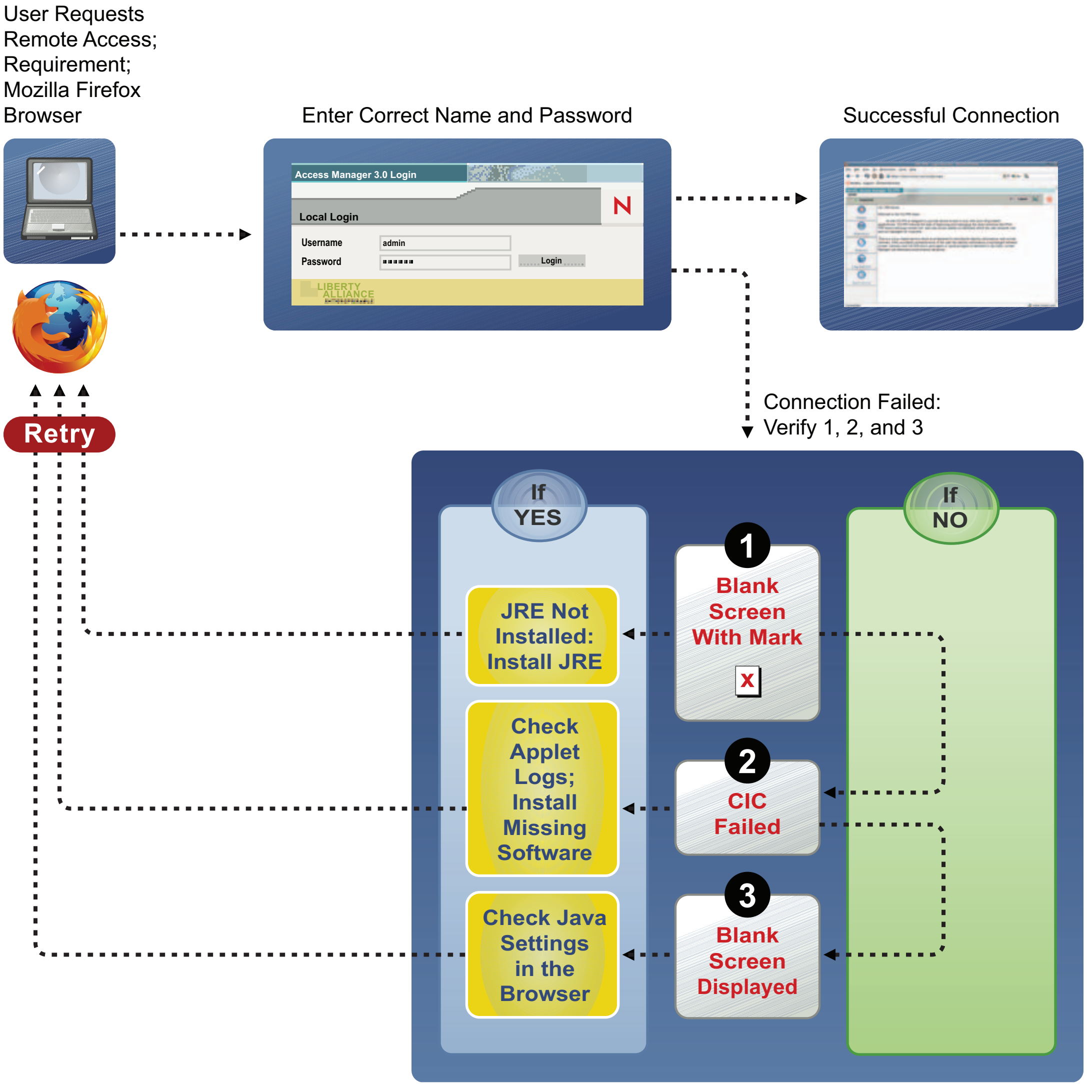
This section provides various troubleshooting scenarios that you might encounter while configuring SSL VPN.

41.1 Connecting Successfully to the Server

You can access the protected resources that are using SSL VPN by authenticating to the proxy server. The proxy server loads the SSL VPN client on your browser. The following sections describe some of the problems that clients might encounter:

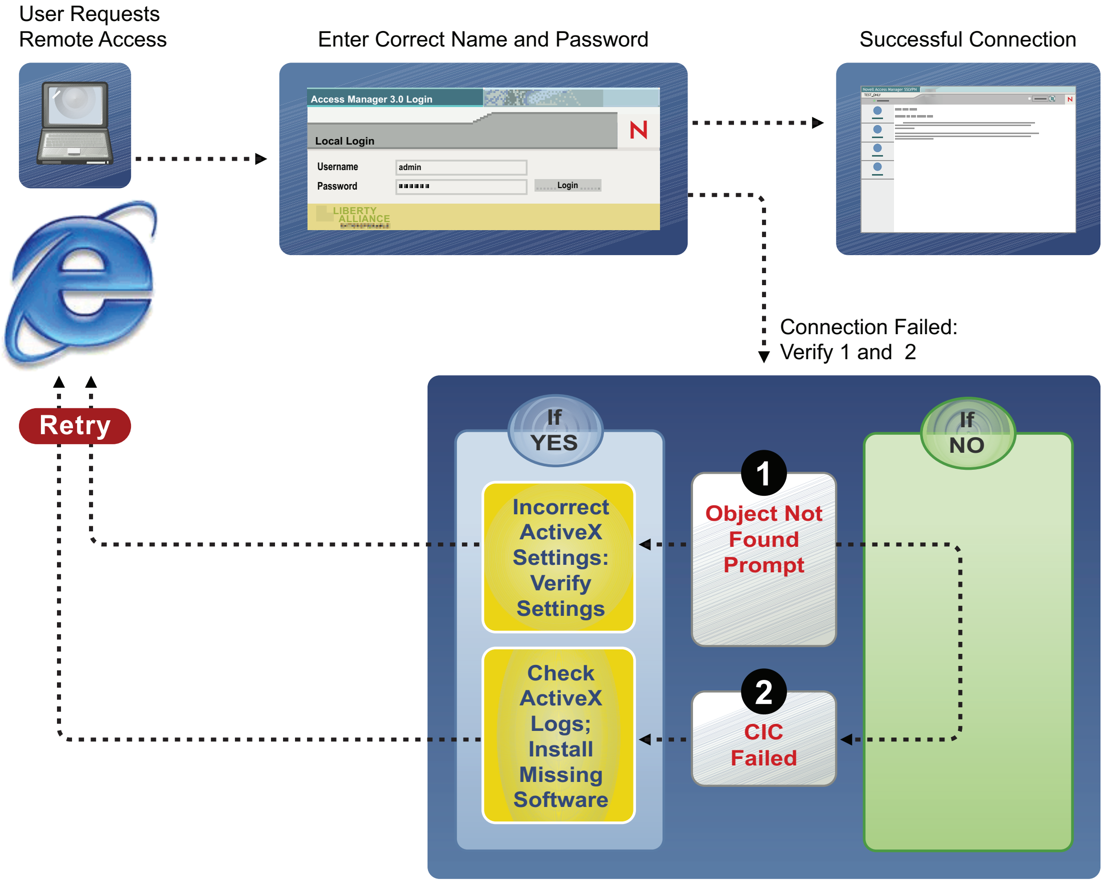
41.1.1 Connection Problems with Mozilla Firefox

Figure 41-1 *Using Mozilla Firefox to Connect to the SSL VPN Server*


User Requests Remote Access; Requirement; Mozilla Firefox Browser
Enter Correct Name and Password
Access Manager 3.0 Login
Local Login
Username
admin
Password
Login
LIBERTY ALLIANCE
Successful Connection
Connection Failed: Verify 1, 2, and 3
Retry
If YES
If NO
1
Blank Screen With Mark
x
2
CIC Failed
3
Blank Screen Displayed
JRE Not Installed: Install JRE
Check Applet Logs; Install Missing Software
Check Java Settings in the Browser

41.1.2 Connection Problems with Internet Explorer

Figure 41-2 *Using Internet Explorer to Connect to the SSL VPN Server*



41.2 TFTP Application Does Not Work in the Enterprise Mode

If the TFTP application does not work in the enterprise mode of SSL VPN, make sure you have done the following:

- You have configured a route using the default gateway.
- You are not using source NAT to route packets.

41.3 SSL VPN Not Reporting

If SSL VPN is not reporting, you must verify the status of JCC, SSL VPN server and restart them if they are down. If restarting any of these components do not work, reconfigure SSL VPN. If none of these work, then you must delete and reimport the SSL VPN server. This section has the following information:

- Section 41.3.1, “Verifying and Restarting JCC,” on page 664
- Section 41.3.2, “Verifying and Restarting SSL VPN Server,” on page 664
- Section 41.3.3, “Reconfiguring SSL VPN,” on page 664

- Section 41.3.4, "Deleting and Reimporting SSL VPN Server," on page 664

41.3.1 Verifying and Restarting JCC

To check the status of JCC, enter the following command:

```
/etc/init.d/novell-jcc status.
```

If it is not running, enter the following command to restart jcc:

```
/etc/init.d/novell-jcc restart
```

41.3.2 Verifying and Restarting SSL VPN Server

To verify the status of SSLVPN server, enter the following command:

```
/etc/init.d/novell-sslvpn status
```

If any component is down, stop and start the SSL VPN server using the following commands:

```
novell-sslvpn stop
novell-sslvpn start.
```

41.3.3 Reconfiguring SSL VPN

If JCC and SSL VPN are up and running but the problem persists, reconfigure SSL VPN as follows: verify if tomcat is up and running, enter the following command:

1. Enter the following command to configure SSL VPN:

   ```
   sslvpnc --configure
   ```

   Specify the following information:

   - IP address of the Administration Console
   - Public IP address of SSL VPN server
   - Private IP address of the SSL VPN server

2. Enter the following command to start the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn start
   ```

3. Enter the following command to restart server communications:

   ```
   /etc/init.d/novell-jcc restart
   ```

41.3.4 Deleting and Reimporting SSL VPN Server

If none of the above work, you must delete the SSL VPN server and reimport it to the administration control. Make sure that you have a backup of the `config.xml` and `config.txt` files, before you proceed with the following steps:

1. In Administration Console, click *SSL VPNs*.
2. Select the SSL VPN server that has the problem, then click *Delete*.
3. Install the SSL VPN Gateway on a new server.

Specify the IP address of Administration Console and the public and private addresses of the SSL VPN Gateway during installation.

After installation, the SSL VPN Gateway is imported into the Administrator Console. This gateway does not have the configuration of the old SSL VPN gateway.

4. In Administration Console, select *Novell Access Manager* > *SSL VPN*. Select the newly added SSL VPN server, then click *Delete*.
5. Copy `config.xml` from the backup device to the following path:

   ```
   /etc/opt/novell/sslvpn/
   ```
6. Copy `config.txt` from the backup device to the following path:

   ```
   /var/opt/novell/tomcat4/webapps/sslvpn/WEB-INF/
   ```
7. As a `root` user, enter the following command to stop the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn stop
   ```
8. Enter the following command to configure SSL VPN:

   ```
   sslvpnc --configure
   ```

   Specify the following information:
   - IP address of the Administration Console
   - Public IP address of SSL VPN server
   - Private IP address of the SSL VPN server
9. Enter the following command to start the SSL VPN server:

   ```
   /etc/init.d/novell-sslvpn start
   ```
10. Enter the following command to restart server communications:

    ```
    /etc/init.d/novell-jcc restart
    ```

This imports the new SSL VPN server into the Administration Console with the configuration of the old SSL VPN server. If you had configured multiple private IP addresses for the old SSL VPN server, you can change them in the Administration Console.

41.4 Verifying SSL VPN Components

Use the commands and processes described in the following sections to verify that the SSL VPN components are running:

- Section 41.4.1, "SSL VPN Server," on page 665
- Section 41.4.2, "SSL VPN Linux Client," on page 666
- Section 41.4.3, "SSL VPN Macintosh Client," on page 666
- Section 41.4.4, "SSL VPN Windows Client," on page 666

41.4.1 SSL VPN Server

To verify the function of all the SSL VPN components, use the commands listed in the table below:

Component	Command
Connection Manager	`pgrep connman`

Component	Command
Sock Daemon	`pgrep sockd`
Secure Tunnel	`pgrep stunnel`
OpenVPN	`pgrep openvpn`

41.4.2 SSL VPN Linux Client

Component	Command
Policy Resolver for Kiosk mode	`pgrep polresolver`
Secure Tunnel for Kiosk mode	`pgrep stunnel`
OpenVPN for Enterprise mode	`pgrep openvpn`

41.4.3 SSL VPN Macintosh Client

Component	Command
Policy Resolver for Kiosk mode	`ps -A | grep polresolver | grep -v grep`
`Secure Tunnel for Kiosk mode`	`ps -A | grep stunnel | grep -v grep`
OpenVPN for Enterprise mode	`ps -A | grep openvpn | grep -v grep`

41.4.4 SSL VPN Windows Client

Check if the stunnel and polresolver processes are up and running, if SSL VPN is in Kiosk mode and openvpn, if SSL VPN is in Enterprise mode.

41.5 Issues With Keep Alive

Sometimes you might receive error messages citing issues with keep alive such as `Failed to receive keepalive. Disconnecting`. These errors might occur because of one of the following reasons:

Possible Cause: The Access Gateway might be down.

Action: Make sure that the Access Gateway is up and running.

Possible Cause: The connection to the server is interrupted. The link might be down.

Action: Try and reconnect to the server.

Possible Cause: The SSL VPN service is down.

Action: Make sure that your SSL VPN service is up and running. Verify all the SSL VPN processes. For more information see Section 41.4, “Verifying SSL VPN Components,” on page 665.

41.6 Unable to Contact the SSL VPN Server

In the client browser if you encounter the message `SSLVPN Gateway is in bad state` or the message `SSLVPN Gateway is not available`, verify the following:

- **Error Status:** Check the status at `/var/log/messages`, `/var/log/stunnel.log`, and `/var/log/novell-openvpn.log`.
- **SSL VPN Status:** At the command prompt, enter the following command:

  ```
  /etc/init.d/novell-sslvpn status
  ```
- **Message Log:** Check the `/var/log/messages` file for more information.

41.7 Unable to Get Authentication Headers

If the browser displays the `Unable to Get Authentication Headers` error while accessing the SSL VPN URL, check whether the custom HTTP headers required for SSL VPN are configured and enabled in the Access Gateway. In the Administration Console, click *Access Gateways > [Configuration Link] > [Name of Reverse Proxy] > [Name of SSL VPN Proxy Service] > [Name of SSL VPN Protected Resource] > Identity Injection*.

The SSLVPN_Default policy should be enabled. This policy injects an authentication header and two custom headers (X-SSLVPN-PROXY-SESSION-COOKIE and X-SSLVPN-ROLE).

41.8 The SSL VPN Connection Is Successful But There Is No Data Transfer

Possible Cause: If this issue appears in the Kiosk mode, the private address specified during the server configuration might be incorrect.

Action: Click *SSL VPNs > Edit > Gateway Configuration*, then check the private address configuration. Make sure that this is the IP address of the private interface of SSL VPN server.

Possible Cause: This issue might occur in both Kiosk as well as the Enterprise modes of SSL VPN. If the SSLVPN server is behind a NAT, the external IP address specified during server configuration might be incorrect.

Action: Click *SSL VPNs > Edit > Gateway Configuration*. Make sure that the external IP address is configured to be the IP address of a NAT through which the external user on the Internet can access the SSL VPN server.

Possible Cause: If this issue appears in the Enterprise mode, it could be because NAT configuration is wrong.

Action: In the command prompt enter `iptable -L` to check the configuration details. For more information see, Section 19.2, “Configuring the IP Address, Port, and NAT,” on page 322

Possible Cause: If this issue appears in the Enterprise mode, it could be because the router configuration is wrong.

Action: Check the router configuration. For more information see, Section 19.2, “Configuring the IP Address, Port, and NAT,” on page 322

Possible Cause: If this issue appears in the Enterprise mode, TUN interface may be down.

Action: In the command prompt, enter `ifconfig` to check if the TUN0 interface is down. If it is down, enter the `etc/init.d/novell-sslvpn restart` command to restart the SSL VPN services.

If you are using a 64-bit machine and have changed the TUN interface, check if that interface is up. If it is down, enter the `etc/init.d/novell-sslvpn restart` command to restart the SSL VPN services.

41.9 Unable to Connect to the SSL VPN Gateway

Possible Cause: A forward proxy is enabled in Internet Explorer.

Action: In the Administration Console, select *Access Manager > Access Gateways > Edit > Reverse Proxy > Proxy List > Path-Based Multi-Homing > HTTP Options*. Select the *Allow Pages to Be Cached by the Browser* check box.

41.10 Multiple Instances of SSL VPN Running

If you get this error while trying to connect to SSL VPN, this could be because there was an improper log out in the previous session and some of the processes did not close properly. Verify if any of the SSL VPN processes are running. For more information on how to verify, see Section 41.4, "Verifying SSL VPN Components," on page 665.

If this error occurs, manually kill the process if you are an admin or a root user of the machine. If you are a non-admin or non-root user of the machine, then restart the machine.

Using the Log Files for Troubleshooting

42

The following sections describe the logging features available in Access Manager and provide information on how you can use them for troubleshooting problems:

- Section 42.1, “Enabling Logging,” on page 669
- Section 42.2, “Understanding Log Format,” on page 672
- Section 42.3, “Sample Authentication Traces,” on page 676

For information about policy tracing, see Section 39.2, “Understanding Policy Evaluation Traces,” on page 600.

42.1 Enabling Logging

Each Access Manager device has configuration options for logging:

Identity Server: Logging is turned off and must be enabled. When you enable Identity Server logging, you also enable logging for the embedded service providers that are configured to use the Identity Server for authentication. For configuration information, see Section 32.2, “Configuring Identity Server Logging,” on page 516.

Embedded Service Providers: Each Access Manager device has an embedded service provider that communicates with the Identity Server. Its log level is controlled by configuring Identity Server logging.

NetWare Access Gateway: Most of the logging available for the NetWare Access Gateway is for its embedded service provider. The log level of this subcomponent is controlled with the Identity Server logging configuration. The logging specific to the NetWare Access Gateway is not configurable, and the NetWare Access Gateway messages are sent to the logger screen.

Linux Access Gateway: A log notice level of logging is enabled by default. You can change the level from the command line interface. For information, see “Linux Access Gateway Logs” on page 669.

42.1.1 Linux Access Gateway Logs

This section contains the following information about the Linux Access Gateway logs:

- “Configuring Log Levels” on page 670
- “Interpreting Log Messages” on page 670
- “Configuring Logging of SOAP Messages and HTTP Headers” on page 672

Configuring Log Levels

You can use the following procedure to set the level of information logged to the `ics_dyn.log` file in the `/var/log` directory.

1 At the command prompt, enter the following command:

```
nash
```

2 At the `nash` shell prompt, enter the following command:

```
configure .current
```

3 To change the log level, enter the following command:

```
log-conf log-level <log level>
```

Replace <*log level*> with the new log level that you want to set.

Level	Description
LOG_EMERG	Sends only messages that render the system unusable, if they are not resolved.
LOG_ALERT	Sends only messages that require immediate action.
LOG_CRIT	Sends only messages about critical situations.
LOG_ERR	Sends warning messages about recoverable errors.
LOG_WARNING	Sends warning messages.
LOG_NOTICE	Sends the service configuration logs information about the status of a service.
LOG_INFO	Sends informational messages such as requests sent to Web servers and the results of authentication requests.
LOG_DEBUG	Sends debug messages.

When you run the `/etc/init.d/novell-vmc start` command, the default log level is set to LOG_NOTICE. You can change the log level to any level from LOG_EMERG to LOG_INFO.

4 To apply changes, enter the following command:

```
apply
```

5 To exit from the configuration mode, enter the following command:

```
exit
```

6 To exit from the nash shell, enter the following command:

```
exit
```

Interpreting Log Messages

In Linux Access Gateway, the entries in the `ics_dyn.log` file have the following format:

```
<time-date-stamp> <hostname> : <AM#event-code> : <AMDEVICE#device-
id> : <AMAUTHID#auth-id> : <AMEVENTID#event-id> :<supplementary log
entry data and text>
```

A sample log message is given below:

```
Aug  3 14:35:41 c1h : AM#504503000: AMDEVICEID#ag-0BDF41AAC4CDCBE5 :
AMAUTHID#0: AMEVENTID#74: Process request 1 'www.lag-202.com' '/
AGLogout' [192.10.100.111:38091 -> 192.10.106.2:80]
```

The fifth and sixth digits in the <AMEVENTID#event-id> refer to the Linux Access Gateway components. The following table list the numbers and the components which they denote.

Table 42-1 *Linux Access Gateway Components*

Number	Component
01	If the fifth and sixth digit are 01, then, it represents the Multi-Homing component.
02	If the number is 02, then it represents the Service Manager component.
03	If the number is 03, then it represents the Request Processing component.
04	If the number is 04, then it represents the Authentication component.
05	If the number is 05, then it represents the Authorization component.
06	If the number is 06, then it represents the Identity Injection component.
07	If the number is 07, then it represents the Form Fill component.
08	If the number is 08, then it represents the Caching component.
09	If the number is 09, then it represents the Response Processing component.
11	If the number is 10, then it represents the Rewriting component.
12	If the number is 11, then it represents the Soap Channel component.
14	If the number is 12, then it represents the VM component.
15	If the number is 15, then it represents the Connection Manager component.
16	If the number is 16, then it represents the VXE component.
17	If the number is 17, then it represents the DataStream component.

For more information on the log format, see Section 42.2, “Understanding Log Format,” on page 672.

Configuring Logging of SOAP Messages and HTTP Headers

1. At the command prompt, enter the following command:
   ```
   nash
   ```
2. To enter the configuration mode, enter the following command:
   ```
   configure .current
   ```
3. Enter one of the following commands to configure logging:

Command	Purpose
`log-conf debug-soap-messages enable`	Logs all the SOAP messages between the Linux Access Gateway and the embedded service provider to the `/var/log/lagsoapmessages` file.
`log-conf no debug-soap-messages enable`	Disables the logging of SOAP messages between the Linux Access Gateway and the Enterprise Server.
`log-conf debug-http-headers enable`	Logs all the HTTP headers between the browsers and the Linux Access Gateway and between the Linux Access Gateway and the Web servers to the `/var/log/laghttpheaders` file.
`log-conf no debug-http-headers enable`	Disables the logging of HTTP headers to the `/var/log/laghttpheaders` file.

4. To apply changes, enter the following command:
   ```
   apply
   ```
5. To exit from the configuration mode, enter the following command:
   ```
    exit
   ```
6. To exit from the nash shell, enter the following command:
   ```
    exit
   ```

42.2 Understanding Log Format

Access Manager does not have a fixed format for file log entries. However, to facilitate the use of non-interactive stream-oriented editors such as sgrep, sed, awk, and grep and to improve log entry readability, the log entries in the `catalina.out` files use some standard elements. These entries use the beginning and ending log entry tags and the log entry correlation tags. The data portion of log entries is the most flexible part. A log entry has the following fields:

```
<amLogEntry> [\n]
   time-date-stamp
   [log preamble]:
   AM#event-code:
   AMDEVICE#device-id:
```

```
   AMAUTHID#auth-id:
   AMEVENTID#event-id:
   [..additional correlating information][\n]
   [supplementary log entry data and text ... \n]
</amLogEntry> [\n]
```

Most log entries do not use the optional line breaks ([\n]). Notice that the time-date-stamp, the log preamble, the correlation tags, and optional additional correlating information are on the same line so that stream-oriented editors that use only one line (such as grep) can be used to locate log entries that are related. The following entry is a typical entry that is logged when a user has initiated a login sequence.

```
<amLogEntry> 2007-06-08T21:06:25Z INFO NIDS Application: AM#500105014:
AMDEVICEID#9921459858EAAC29: AMAUTHID#BB11C254B7521B5E836D8703826287
AF:  Attempting to authenticate user cn=jwilson,o=novell with provided
credentials. </amLogEntry>
```

***Table 42-2** Fields in a Log Entry*

Field	Description
Beginning, ending tags	The `<amLogEntry>` and `</amLogEntry>` tags mark the beginning and the end of a log entry. These tags allow stream-oriented editors to extract log entries for processing.
Time-date-stamp tag	The date and time is specified in the W3C profile format of ISO 8061. It has the following fields: year-month-day-T-hour-minutes-seconds-time zone. The Z value for the time zone indicates that the time is specified in UTC.
Log preamble	This information is optional, and usually consists of a string indicating the logging level (such as warning, informational, or debug) and a string identifying the type of module making the entry. In the example log entry, the preamble has a log level and a module identifier and contains the following strings: `INFO NIDS Application`:
Correlation tags	The correlation tags uniquely identify the event, the device that produced the event, and the user who requested the action. The example log entry contains the following correlation tags: `AM#500105014: AMDEVICEID#9921459858EAAC29: AMAUTHID#BB11C254B7521B5E836D8703826287AF:` For more information, see Section 42.2.1, “Understanding the Correlation Tags in the Log Files,” on page 674.
Additional correlation information	This information is optional, and contains correlation tags and data unique to a specific type of trace. For example, a policy evaluation trace created by the embedded service provider contains the following additional tags: ◆ NXPESID#value ◆ POLICYID#value The example log entry does not contain any additional correlation information. For a log entry that does, see Section 39.2.4, “Identity Injection Traces,” on page 610.

Field	Description
Supplementary information	This information is optional, and contains information that is specific to the log entry. It can be as simple as an informational string, such as the string in the example log entry: `Attempting to authenticate user cn=jwilson,o=novell with provided credentials.` The supplementary information can have a very specific format. For an example and explanation of the policy trace information, see Section 39.2, "Understanding Policy Evaluation Traces," on page 600.

42.2.1 Understanding the Correlation Tags in the Log Files

There is no fixed field format for log file entries. However, because most requests handled by Access Manager are processed by multiple Access Manager components, there is a mechanism defined that facilitates the correlation of log entries for a single Access Manager request in the various component log files. A correlation tag has the following general format:

```
<tag name>#<tag value>:
```

The `<tag name>` is a fixed value, defined in the Format column of Table 42-3. It is always terminated by the # character. The `<tag value>` begins immediately following the # character and is always terminated by the : character. The `<tag value>` is not a fixed value, but a uniquely assigned value to identify an event, a user, or a transaction. Table 42-3 lists the defined correlation tags:

Table 42-3 *Correlation Tags*

Type	Format	Description
Event code	`AM#<Event-Code>:`	An event number defined in Event Codes (http://www.novell.com/documentation/novellaccessmanager/eventcodes/data/bookinfo.html). This tag is included in all log entries that record an event and in all events that are presented to the user as an informational or error page.

Type	Format	Description
User ID	`AMAUTHID#<ID>:`	An authentication identifier that the Identity Server or the embedded service provider assigns to each authenticated user. This tag is included in all entries that pertain to a request made by an authenticated user. Currently the Identity Server and the embedded service provider (ESP) assign different authentication IDs. When correlating the flow of events between the Identity Server and the ESP for an authentication sequence, you can use the event code of the authentication events and find the artifact that the ESP and the Identity Server exchange. In the `catalina.out` file of the Identity Server, search for `AM#500105018` events. This is the event that sends the artifact to the ESP. Search for a corresponding artifact in the Access Gateway log. Events `AM#500105020` and `AM#500105021` contain the artifact value.
Device ID	`AMDEVICE#<ID>`	An identifier that uniquely identifies the Access Manager device that is generating the log entry. You can view the identifier that is assigned to each device on the General Logging page in the Administration Console (click *Access Gateways* > *Auditing* > *General Logging*). The ID begins with a prefix that identifies the type of device such as idp for Identity Server, ag for an Access Gateway, and idp-esp for the embedded service provider of the device. The prefix is followed by a 16-digit hexadecimal number. In log entries, the idp prefix is not recorded. For example, the General Logging page displays `idp-AA257DA77ED48DB0` for the ID of the Identity Server, but in the `catalina.out` file, the value is `AMDEVICE#AA257DA77ED48DB0`.

Type	Format	Description
Transaction ID	`AMEVENTID#<ID>:`	An identifier assigned to each Access Manager or system administration transaction. Access Manager transactions are such actions as authenticating a user, processing a request for access to a resource, and federating an identity. If a user requests access to multiple resources, each request is given a separate transaction ID. When the Access Gateway evaluates a policy for a protected resource page and the page contains links, the policy is evaluated for each link, and each of these evaluations generates a new transaction ID. System administration transactions are such actions as importing a device, deleting a device, stopping or starting a device, and configuring or modifying the configuration of a device.

42.2.2 Sample Scenario

The following scenario illustrates how these tags can be used. A user receives an error page indicating he or she has been refused access to a protected resource. The error page contains an event code. The user contacts the system administrator and reports the event code contained in the message. The code displayed to the user includes both an event number and an identifier indicating the device detecting the error, for example, `300101023-92E1B234`. The `300101023` value is the event number and `92E1B234` is the device identifier. The device identifier is the number assigned to the Access Manager device reporting the error. You can make a textual search of log entries using the tags and values `AM#300101023:` and `AMDEVICEID#92E1B234:` to locate candidate log entries of the target Access Manager transaction flow. When the desired log entry is found, the `AMEVENTID#` tag and value and the `AMAUTHID# tag` (assuming the user has been authenticated) from the log entry can be used to locate all other log entries pertaining to the user in the context of the transaction.

42.3 Sample Authentication Traces

An authentication trace is logged to the `catalina.out` file of the Identity Server that authenticates the user. If the Access Gateway initiates the authentication because of a user request to a protected resource, the embedded service provider log file of the Access Gateway also contains entries for the authentication sequence. Identity Server logging must be enabled to produce authentication traces (see Section 32.2, "Configuring Identity Server Logging," on page 516).

This section describes the following types of authentication traces:

- Section 42.3.1, "Direct Authentication Request to the Identity Server," on page 676
- Section 42.3.2, "Protected Resource Authentication Trace," on page 679

42.3.1 Direct Authentication Request to the Identity Server

The following trace is an example of a user logging directly into the Identity Server to access the End User Portal. The log entries have been modified to add numbers, so that they can be described.

```
1. <amLogEntry> 2007-06-14T17:14:30Z INFO NIDS Application:
AM#500105015: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Processing login request
with TARGET = http://10.10.15.19:8080/nidp/app, saved TARGET = . </
amLogEntry>

2. <amLogEntry> 2007-06-14T17:14:30Z INFO NIDS Application:
AM#500105009: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Executing contract Name/
Password - Form. </amLogEntry>

3. <amLogEntry> 2007-06-14T17:14:30Z INFO NIDS Application:
AM#500105010: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Contract Name/Password -
Form requires additional interaction. </amLogEntry>

4. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500105015: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Processing login request
with TARGET = http://10.10.15.19:8080/nidp/app, saved TARGET = http://
10.10.15.19:8080/nidp/app. </amLogEntry>

5. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500105009: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Executing contract Name/
Password - Form. </amLogEntry>

6. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500105014: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Attempting to authenticate
user cn=bcf,o=novell with provided credentials. </amLogEntry>

7. <amLogEntry> 2007-06-14T17:14:39Z WARNING NIDS Application: Event
Id: 3014666, Target: cn=bcf,o=novell, Sub-Target:
F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 1: Local, Note 2: This Identity
Provider, Note 3: name/password/uri, Numeric 1: 0 </amLogEntry>

8. <amLogEntry> 2007-06-14T17:14:39Z WARNING NIDS Application: Event
Id: 3015456, Note 1: F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 2:
Manager, Note 3: Document=(ou=xpemlPEP,ou=mastercdn,ou=Content
PublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou
=accessManagerContainer,o=novell:romaContentCollectionXMLDoc),Policy=(
Manager),Rule=(1::RuleID_1181251958207),Action=(AddRole::ActionID_1181
252224665), Numeric 1: 0 </amLogEntry>

9. <amLogEntry> 2007-06-14T17:14:39Z WARNING NIDS Application: Event
Id: 3015456, Note 1: F35A3C7AD7F2EEDEB3D17F99EC3F39D1, Note 2:
authenticated, Note 3: system-generated-action, Numeric 1: 0 </
amLogEntry>

10. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500199050: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: IDP RolesPep.evaluate(),
policy trace:
   ~~RL~1~~~~Rule Count: 1~~Success(67)
   ~~RU~RuleID_1181251958207~Manager~DNF~~1:1~~Success(67)
   ~~CS~1~~ANDs~~1~~True(69)
   ~~CO~1~LdapGroup(6645):no-param:hidden-value:~ldap-group-is-member-
of~SelectedLdapGroup(66455):hidden-param:hidden-value:~~~True(69)
   ~~PA~ActionID_1181252224665~~AddRole~Manager~~~Success(0)
   ~~PC~ActionID_1181252224665~~Document=(ou=xpemlPEP,ou=mastercdn,
ou=ContentPublisherContainer,ou=Partition,ou=PartitionsContainer,ou=VC
DN_Root,ou=accessManagerContainer,o=novell:romaContentCollectionXMLDoc
),Policy=(Manager),Rule=(1::RuleID_1181251958207),Action=(AddRole::Act
ionID_1181252224665)~AdditionalRole(6601):unknown():Manager:~~~Success
(0)
 </amLogEntry>

11. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500105013: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: Authenticated user
cn=bcf,o=novell in User Store Local Directory with roles
Manager,authenticated. </amLogEntry>

12. <amLogEntry> 2007-06-14T17:14:39Z INFO NIDS Application:
AM#500105017: AMDEVICEID#9921459858EAAC29:
AMAUTHID#F35A3C7AD7F2EEDEB3D17F99EC3F39D1: nLogin succeeded,
redirecting to http://10.10.15.19:8080/nidp/app. </amLogEntry>
```

Table 42-4 *Log Entry Descriptions for an Authentication Trace from an Identity Server*

Entry	Description
1	Indicates that a login request is in process. This is the first entry for a login request. The requester, even though login has not been successful, is assigned an authentication ID. You can use this ID to find the log entries related to this user. The entry also specifies the URL of the requested resource, in this case the /nidp/app resource called the End User Portal. The saved TARGET message does not contain a value, so this step will be repeated.
2	Specifies the contract that is being used to perform the login.
3	Indicates that the contract requires interaction with the user.
4	Indicates that the a login request is in process. The saved TARGET message contains a value, so the required information has been gathered to start the authentication request. The AM# correlation tag is `AM#500105015`, which is the same value as the first log entry.
5	Indicates that an exchange is occurring between the client and the Identity Server to obtain the required credentials. Each contract requires a different exchange. The `AM#` correlation tag is AM#500105009, which is the same value as the second log entry.
6	Provides the DN of the user attempting the log in and indicates that the user's credentials are being sent to the LDAP server for verification.
7	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file. This event contains information about who is logging in and the contract that is being used.
8	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file. This event contains information about the Manager policy that is evaluated during login.

Entry	Description
9	Provides information about an auditing event. If you have not enabled auditing or you have not selected the login events, this entry does not appear in your log file.
10	Contains the entry for processing a Role policy. When a user logs in, all Role policies are evaluated and the user is assigned any roles that the user has the qualifications for. For more information, see Section 39.2, “Understanding Policy Evaluation Traces,” on page 600.
11	Contains a summary of who logged in from which user store and the names of the Role policies that successfully assigned roles to the user.
12	Contains the final results of the login, with the URL that the request is redirected to.

42.3.2 Protected Resource Authentication Trace

When a protected resource is configured to require authentication, both the Identity Server and the embedded service provider of the Access Gateway (or J2EE Agent) generate log entries for the process. The following sections explain how to correlate the entries from the logs.

- “Entries from an Identity Server Log” on page 679
- “Entries from an Access Gateway Log” on page 680
- “Correlating the Log Entries between the Identity Server and the Access Gateway” on page 681

Entries from an Identity Server Log

```
<amLogEntry> 2007-07-31T17:36:39Z INFO NIDS Application: AM#500105016:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Processing login resulting
from Service Provider authentication request. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:39Z INFO NIDS Application: AM#500105009:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Executing contract Name/
Password - Form. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:39Z INFO NIDS Application: AM#500105010:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Contract Name/Password -
Form requires additional interaction. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105016:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Processing login resulting
from Service Provider authentication request. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105009:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Executing contract Name/
Password - Form. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105014:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Attempting to authenticate
```

```
user cn=admin,o=novell with provided credentials. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105012:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Authenticated user
cn=admin,o=novell in User Store Internal with no roles. </amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105018:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#83778AE09DCA5A35B57842D754A60D67: Responding to AuthnRequest
with artifact AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/qBNool8WkZiTCt7N7Jx
</amLogEntry>

<amLogEntry> 2007-07-31T17:36:49Z INFO NIDS Application: AM#500105019:
AMDEVICEID#AA257DA77ED48DB0:
AMAUTHID#C2D8D52704918AF2D5D62F6EDC2FFAC6: Sending AuthnResponse in
response to artifact AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/
qBNool8WkZiTCt7N7Jx </amLogEntry>
```

Entries from an Access Gateway Log

```
<amLogEntry> 2007-07-31T17:35:05Z INFO NIDS Application: AM#500105005:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Processing proxy request
for login using contract name/password/uri and return url http://
jwilson.provo.novell.com/ </amLogEntry>

<amLogEntry> 2007-07-31T17:35:05Z INFO NIDS Application: AM#500105015:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Processing login request
with TARGET = http://jwilson.provo.novell.com/, saved TARGET = . </
amLogEntry>

<amLogEntry> 2007-07-31T17:35:05Z INFO NIDS Application: AM#500105009:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Executing contract IDP
Select. </amLogEntry>

<amLogEntry> 2007-07-31T17:35:05Z INFO NIDS Application: AM#500105010:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Contract IDP Select
requires additional interaction. </amLogEntry>

<amLogEntry> 2007-07-31T17:35:15Z INFO NIDS Application: AM#500105020:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Received and processing
artifact from IDP - AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/
qBNool8WkZiTCt7N7Jx </amLogEntry>

<amLogEntry> 2007-07-31T17:35:15Z INFO NIDS Application: AM#500105021:
AMDEVICEID#esp-2FA73CE1A376FD91:
AMAUTHID#C6D119FD93EEBBEBEC50BEB27B9E2832: Sending artifact
AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/qBNool8WkZiTCt7N7Jx to URL http:/
/jwilson1.provo.novell.com:8080/nidp/idff/soap at IDP </amLogEntry>
```

Correlating the Log Entries between the Identity Server and the Access Gateway

You can tell that these two trace sequence are for the same authentication request because the artifact (`AAMoz+rm2jQjDSHjea8U9zm3Td/U2ax0YZCo/qBNool8WkZiTCt7N7Jx`) that is exchanged is the same. You can use the `AMAUTHID` in each file to search for other requests that this user has made.

To associate a distinguished name with the `AMAUTHID`, use the `catalina.out` file of the Identity Server. Event `AM#500105014` contains the DN of the user.

Troubleshooting XML Validation Errors

43

A XML validation error is often ignored because the returning message does not appear to be serious. However, closer inspection of the Linux Access Gateway shows that none of the changes have been applied. When a change is applied by using the UI, the system writes the configuration to the configuration store on the Administration Console, as well as to the `/var/novell/cfgdb/vcdn/config.xml` file on the Linux Access Gateway. If this file passes the schema checks on the Linux Access Gateway, the `/var/novell/cfgdb/.current/config.xml` file is updated with the configuration.

This is the file that the Linux Access Gateway reads when it loads or refreshes. If the `config.xml` file from `/var/novell/cfgdb/vcdn/` and `/var/novell/cfgdb/.current` are not in sync, then all changes you defined have not been applied to the Linux Access Gateway.

You need to pay attention to XML validation errors and identify the key steps required to solve such problems. There are two main scenarios that are discussed in this section:

- Section 43.1, "Modifying a Configuration That References a Removed Object," on page 683
- Section 43.2, "Configuration UI Writes Incorrect Information to the Local Configuration Store," on page 685

43.1 Modifying a Configuration That References a Removed Object

One scenario that causes XML validation errors occurs when a configuration references an object that has been removed. For example, a custom authentication contract was created and assigned to a protected resource. The contract was manually deleted from the Identity Server configuration, but the Access Gateway protected resource still references it, even though it is not displayed in the user interface. After you identify the missing link, you can use the Access Manager interface to work around the problem.

Troubleshooting Steps

1 Search the `/opt/novell/devman/share/logs/app_sc.0.log` file on the Administration Console server for #200904025: Error - XML VALIDATION FAILED.

After you find the entry, work backwards to identify the start of the Java exception. Locate the problem strings or entry from the configuration, such as the following string `authprocedure_NEIL___Name_Password___Form` found in the following entry:

```
871(D)Wed May 23 15:45:06 BST
2007(L)webui.sc(T)26(C)com.volera.vcdn.webui.sc.dispatcher.ConfigW
orkDispatcher(M)A(E)org.jdom.input.JDOMParseException: Error on
line 1120: cvc-id.1: There is no ID/IDREF binding for IDREF
'authprocedure_NEIL___Name_Password___Form'.
at org.jdom.input.SAXBuilder.build(SAXBuilder.java:468)
at org.jdom.input.SAXBuilder.build(SAXBuilder.java:770)
at com.volera.vcdn.platform.util.XmlUtil.validateXML(y:3304)
at
com.volera.vcdn.webui.sc.dispatcher.ConfigWorkDispatcher.A(y:793)
at
com.volera.vcdn.webui.sc.dispatcher.ConfigWorkDispatcher.do_device
Con
fig(y:648)
:
:
:
at
org.apache.coyote.http11.Http11Processor.process(Http11Processor.j
ava :799)
at
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.pr
oce ssConnection(Http11Protocol.java:705)
at
org.apache.tomcat.util.net.TcpWorkerThread.runIt(PoolTcpEndpoint.j
ava :577)
at
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(Thre
adPool.java:683)
at java.lang.Thread.run(Thread.java:534)
(Msg)<amLogEntry> 2007-05-23T15:45:06Z ERROR DeviceManager:
AM#200904025: Error
- XML VALIDATION FAILED. PLEASE CHECK APP_SC LOG </amLogEntry>
```

2 On the Linux Access Gateway, change to the `/var/novell/cfgdb/vcdn` directory and open the `config.xml` file. Search for the problem string and the corresponding protected resource.

The example below shows that the problem string is tied to the `ProtectedResourceID_svhttp_mylag_iMon_root` resource. This maps to the HTTP reverse proxy called `mylag`, the service called `iMon` and the protected resource called `root`.

```
----- snippet from problem area of config.xml ------
<ProtectedResource Name="root" Enable="1" Description=""
LastModified="116973455
5995" LastModifiedBy="cn=admin,o=novell"
UserInterfaceID="ProtectedResourceID_sv
http_mylag_iMon_root"
ProtectedResourceID="ProtectedResourceID_svhttp_mylag_iMon
_root">

        <URLPathList LastModified="4294967295"
LastModifiedBy="String">

                <URLPath URLPath="/*" UserInterfaceID="/*"/>

        </URLPathList>
        <PolicyEnforcementList LastModified="1168947602067"
schemaVersion="1.34"
 LastModifiedBy="cn=admin,o=novell"
RuleCombiningAlgorithm="DenyOverridesWithPri
ority">

                <PolicyRef ElementRefType="ExternalWithIDRef"
ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherContai
ner,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessMana
gerContainer,o=novell:romaContentCollectionXMLDoc"
UserInterfaceID="PolicyID_xpemlPEP_AGFormFill_1168947167634"
ExternalElementRef="PolicyID_xpemlPEP_AGFormFill_1168947167634"/>

        </PolicyEnforcementList>

        <AuthenticationProcedureRef
AuthProcedureIDRef="authprocedure_NEIL___Name_Password___Form"/>

</ProtectedResource>

----- end of snippet from problem area of config.xml ------
```

Looking at the `AuthenticationProcedureRef` variable, which points to the contract assigned to the protected resource, you can see that the `authprocedure_NEIL___Name_Password___Form` contract is assigned to it.

However, when you look at the Linux Access Gateway configuration in the Administration Console, you can see that the assigned contract is *[None]*, which is not the contract shown in the example. Changing it to another contract name, then applying the change, then setting the contract back to *[None]* clears the problem entry. This gets the setup going again with no XML validation errors.

In this example, these was a custom contract assigned to the protected resource. This custom contract had been removed from the Identity Server's list of contracts, and the cleanup was never done properly on the Linux Access Gateway.

43.2 Configuration UI Writes Incorrect Information to the Local Configuration Store

In this scenario, you apply the same change twice in quick succession, and the information written to the configuration store is invalid. Subsequent schema checks detect this invalid configuration and return an XML validation error. This scenario is more complex because it involves changing the configuration store on the Administration Console.

Troubleshooting Steps

1. On the Administration Console, search the `/opt/novell/devman/share/logs/app_sc.0.log` file for `#200904025: Error - XML VALIDATION FAILED`.

   After you find the entry, work backwards to identify the start of the Java exception. From this, locate the problem strings or entry from the configuration, such as `ProtectedResourceID_svhttp_sjh_portal_sjh_portal_1179933619340`. This message also indicates that a defined protected resource might not be unique. The configuration shows that before the Java exception, there is not enough information to narrow down the problem, so more troubleshooting is required.

The following is a snippet from the problem area of `app_sc.0.log` file that indicates that there are multiple occurrences of a protected resource.

```
Caused by: org.xml.sax.SAXParseException: cvc-id.2: There are
multiple occurrences of ID value
'ProtectedResourceID_svhttp_sjh_portal_sjh_portal_1179933619340'.
at
org.apache.xerces.util.ErrorHandlerWrapper.createSAXParseException
(Unknown Source)
at org.apache.xerces.util.ErrorHandlerWrapper.error(Unknown
Source)
at org.apache.xerces.parsers.XML11Configuration.parse(Unknown
Source)
at org.apache.xerces.parsers.XMLParser.parse(Unknown Source)
at org.apache.xerces.parsers.AbstractSAXParser.parse(Unknown
Source)
at org.jdom.input.SAXBuilder.build(SAXBuilder.java:453)
at org.jdom.input.SAXBuilder.build(SAXBuilder.java:770)
at com.volera.vcdn.platform.util.XmlUtil.validateXML(y:3304)
at
com.volera.vcdn.webui.sc.dispatcher.ConfigWorkDispatcher.A(y:793)
at
com.volera.vcdn.webui.sc.dispatcher.ConfigWorkDispatcher.do_device
config(y:648)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorIm
pl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAc
cessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:324)
at
com.volera.vcdn.webui.sc.dispatcher.DefaultDispatcher.invoke(y:469
)
at
com.volera.vcdn.webui.sc.dispatcher.DefaultDispatcher.processReque
st(y:1732)
at
com.volera.roma.app.handler.DispatcherHandler.processRequest(y:316
8)
at com.volera.roma.servlet.GenericController.doPost(y:53)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:716)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:809)
at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(A
pplicationFilterChain.java:200)
at
org.apache.catalina.core.ApplicationFilterChain.doFilter(Applicati
onFilterChain.java:146)
at
```

```
org.apache.catalina.core.StandardPipeline$StandardPipelineValveCon
text.invokeNext(StandardPipeline.java:594)
at
com.novell.accessmanager.tomcat.SynchronizationValve.invoke(y:297)
at
org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.
java:433)
at
org.apache.catalina.core.ContainerBase.invoke(ContainerBase.java:9
48)
at
org.apache.coyote.tomcat4.CoyoteAdapter.service(CoyoteAdapter.java
:152)
at
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.pr
ocessConnection(Http11Protocol.java:705)
at
org.apache.tomcat.util.threads.ThreadPool$ControlRunnable.run(Thre
adPool.java:683)
at java.lang.Thread.run(Thread.java:534)
(Msg)<amLogEntry> 2007-05-23T13:22:15Z ERROR DeviceManager:
AM#200904025: Error - XML VALIDATION FAILED. PLEASE CHECK APP_SC
LOG </amLogEntry>
```

2 Confirm that the change has not been applied at the Linux Access Gateway. To do this, use the following steps:

  2a Enable the most verbose level of logging in the `/etc/laglogs.conf` file: `log_level=LOG_DEBUG`. See "Configuring Log Levels" on page 670.

  2b Restart the vmc services by the following command:

  ```
  /etc/init.d/novell-vmc restart
  ```

  2c Search for in-memory errors in the `ics_dyn` log file. When these errors are displayed, the working Linux Access Gateway configuration has not been updated with the latest changes.

  2d Identify the protected resource with these issues. In the following case, the protected resource is the same, so you must look at the `config.xml` file and search for this specific protected resource. For example:

  ```
  May 23 13:22:14 chw-amtlag1-176 : 404502  0: 7168: 0: 0:
  VcpConfiguration::reconfigure starting AafLog
  May 23 13:22:14 chw-amtlag1-176 : 404502  0: 7168: 0: 0:
  VcpConfiguration::reconfigure finished
  Error at file "in-memory", line 328, column 306
     Message: Datatype error: Type:InvalidDatatypeValueException,
  Message:ID
  'ProtectedResourceID_svhttp_sjh_portal_sjh_portal_1179933619340
  ' is not unique.
  ERROR: Error retrieving config.xml: No data available
  ```

3 Search for the preceding string in the `/var/novell/cfgdb/vcdn/config.xml` file. Doing this, you should see the following type of information:

```
<ProtectedResourceList>
```

```
<ProtectedResource Name="sjh_redirect" Enable="1"
      Description="" LastModified="1179934022767"

LastModifiedBy="cn=admin,o=novell"UserInterfaceID="ProtectedResour
ceID_svhttp_sjh_portal_sjh_portal_1179933619340"
ProtectedResourceID="ProtectedResourceID_svhttp_sjh_portal_sjh_por
tal_1179933619340">
     <URLPathList LastModified="4294967295"
LastModifiedBy="String">
<URLPath URLPath="/*" UserInterfaceID="/*"/>
      </URLPathList>
      <PolicyEnforcementList LastModified="1179934011081"
schemaVersion="0.1" LastModifiedBy="cn=admin,o=novell"
RuleCombiningAlgorithm="DenyOverridesWithPriority"
IncludedPolicyCategories=""/>
      <AuthenticationProcedureRef
AuthProcedureIDRef="authprocedure_Name_Password___Form"/>
      </ProtectedResource>
   </ProtectedResourceList>
```

and

```
      <ProtectedResourceList LastModified="1179949051828"
LastModifiedBy="cn=admin,o=novell">
      <ProtectedResource Name="sjh_redirect" Enable="1"
Description="" LastModified="1179949051828"
LastModifiedBy="cn=admin,o=novell"
UserInterfaceID="ProtectedResourceID_svhttp_sjh_portal_sjh_portal_
1179933619340"
ProtectedResourceID="ProtectedResourceID_svhttp_sjh_portal_sjh_por
tal_1179933619340">
      <URLPathList LastModified="4294967295"
LastModifiedBy="String">
      <URLPath URLPath="/*" UserInterfaceID="/*"/>
      </URLPathList>
      <PolicyEnforcementList LastModified="1179949047445"
schemaVersion="0.1" LastModifiedBy="cn=admin,o=novell"
RuleCombiningAlgorithm="DenyOverridesWithPriority"
IncludedPolicyCategories="">
      <PolicyRef ElementRefType="ExternalWithIDRef"
ExternalDocRef="ou=xpemlPEP,ou=mastercdn,ou=ContentPublisherContai
ner,ou=Partition,ou=PartitionsContainer,ou=VCDN_Root,ou=accessMana
gerContainer,o=novell:romaContentCollectionXMLDoc"
UserInterfaceID="PolicyID_xpemlPEP_AGAuthorization_1176770874051"
ExternalElementRef="PolicyID_xpemlPEP_AGAuthorization_117677087405
1"/>
      </PolicyEnforcementList>
      <AuthenticationProcedureRef
AuthProcedureIDRef="authprocedure_Name_Password___Form"/>
      </ProtectedResource>
     </ProtectedResourceList>
```

This is the duplicate entry that is causing the problem. You need to clear one of the entries from the configuration. If you clear it from the `/var/novell/cfgdb/vcdn/config.xml` file, then any change applied in the UI rewrites the information to the `config.xml` file.

4 Remove the duplicate entry from the Administration Console server's configuration store. To do this, you need an LDAP browser.

You can download a free Java-based tool from the Internet, for example the LDAP Browser/Editor (http://www-unix.mcs.anl.gov/~gawor/ldap/).

- **4a** Start the LDAP browser, then locate the ag-xxxx that matches the Linux Access Gateway you are having problems with.

  The easiest way is to go to the *Auditing* > *General Logging* tab of the Access Manager Administration Console and identify your Linux Access Gateway ID. This ID corresponds to the first four digits of the ag-xxxx in the LDAP browser.
- **4b** Click the ag-xxxx container. You should see *CurrentConfig* and *WorkingConfig* containers within this Access Gateway container.
- **4c** Select the *CurrentConfig*, then the `RomaAGConfigurationXMLDoc` attribute. Copy and paste the attribute value into any editor. This is the configuration from the LAG.
- **4d** Search for the `RomaAGConfigurationXMLDoc` attribute string and remove the entire section on one of the hits starting with `<ProtectedResourceList>` and ending with `</ProtectedResourceList>`.
- **4e** Select and save the modified text.
- **4f** Paste the saved text into the `RomaAGConfigurationXMLDoc` attribute value.
- **4g** Repeat these steps for the `RomaAGConfigurationXMLDoc` attribute in *WorkingConfig*, and remove the duplicate entry that is causing the XML validation errors.

5 Restart tomcat on the Administration Console.

6 Log in to the Administration Console again. Make a small change to the setup and apply that change, and verify that the XML validation error has disappeared.

Troubleshooting Certificate Issues

44

44.1 Resolving a -1226 PKI Error

When you create a certificate signing request, send it to a third-party issuer to be signed, and receive the server certificate from the third-party issuer, you sometimes receive a -1226 error when you try to import the signed certificate. You receive this error when the issuer does not sent back the trusted roots required to validate the issuer of the server certificate.

Use one of the following options to resolve this issue:

- If the issuer included the trusted root and any intermediate certificates in a separate file or files, specify these files during the import by clicking the + character that allows you to add a trusted root or an intermediate certificate.
- If the issuer did not send you any additional files, you can go to the issuer's Web site, download them, then specify these files during the import by clicking the + character that allows you to add a trusted root or an intermediate certificate.
- You can try importing the certificate into Internet Explorer, which has the trusted roots from all major CAs, then export the certificate with the required chain of trusted roots. See Section 44.1.1, "Using Internet Explorer to Add a Trusted Root Chain," on page 691.

44.1.1 Using Internet Explorer to Add a Trusted Root Chain

The following procedure only works when Internet Explorer contains the trusted root certificate of the issuer of your certificate.

1. In Internet Explorer, click *Tools* > *Internet Options* > *Content* > *Certificates*.
2. Click *Import* and import your server certificate into the *Other People* tab.
3. Click *Other People*, then double click on your certificate.
4. Click *Certification Path*.
   - If the *Certification Path* shows that the certificate is OK, you now have the full certificate chain available for export. Click *OK*, then continue with Step 5.
   - If the *Certification Path* is not OK, you cannot use this method. Click *OK*, then contact your issuer for the certificate chain.
5. Select the certificate, then click *Export* > *Next*.

6. Select *Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)* as the format and select *Include all certificates in the certification path if possible* to include the certificate chain.
7. Click *Next*, then specify a filename and path for the file.
8. Click *Next* > *Finish*.
9. Use this P7B file to import your server certificate into Access Manager.

44.2 Importing an External Certificate Key Pair

The Access Manager Certificate Authority requires that all certificate key pairs in `.pfx` format contain the complete certificate chain. If a key pair was created with multiple CAs and the exported certificate does not contain the complete certificate chain, the file cannot be imported into Access Manager. When you try to import such a certificate, the following error message is displayed:

```
"Error importing certificate key pair: Error: Error: -1403
```

When exporting the certificate key pair, make sure you include all the certificates in the certification path.

To ensure that your certificate contains all the intermediate certificates and contains them in the right order, import the certificate into Internet Explorer or Firefox.

- For Internet Explorer 7, click *Tools* > *Internet Options* > *Content* > *Certificates* > *Personal* > *Import*.
- For Firefox 2, click *Tools* > *Options* > *Advanced* > *Encryption* > *View Certificates* > *Your Certificates* > *Import*.

Make sure the browser contains the public key for all the intermediate CAs. Then select the certificate and export the certificate in `.pfx` format. In Internet Explorer, you must select to include all the certificates in the chain. In Firefox, all the certificates in the chain are automatically included.

If you receive an error when importing the certificate, the error comes from either NICI or PKI. For a description of these error codes, see Novell® Certificate Server Error Codes and Novell International Cryptographic Infrastructure (http://www.novell.com/documentation/nwec/index.html)

44.3 Mutual SSL with X.509 Produces Untrusted Chain Messages

When you set up an X.509 contract for mutual SSL authentication, you must ensure that the Identity Server trust store (NIDP-truststore) contains the trusted root from each CA that has signed the client certificates. If a client has a certificate signed by a CA that is not in the NIDP-truststore, authentication fails.

To add a certificate to the NIDP-truststore:

1. In the Administration Console, click *Access Manager* > *Certificates* > *Trusted Roots* > *NIDP-truststore*.
2. Click either *Add* or *Auto-Import From Server* and follow the prompts.

44.4 Certificate Command Failure

Certificate commands are generated when you upgrade the Administration Console, and you should ensure that they have completed successfully (click *Access Manager* > *Certificates* > *Command Status*).

If a certificate command fails:

1. Note the destination trust store or keystore
2. Click *Auditing* > *Troubleshooting* > *Certificates*.
3. Select the store, then click *Re-push certificates*.

   This pushes all assigned certificates to the store. You can re-push certificates multiple times without causing any problems.

44.5 Can't Log In with Certificate Error Messages

After an upgrade if your users can't log in to access protected resources, and the failure messages contain certificate error messages, you might need to manually push the certificates from the Administration Console to the Access Gateway.

To re-push a certificate:

- For a reverse proxy certificate, go to the Reverse Proxy page, select a different certificate, click *OK*, return to the Reverse Proxy page, select the correct certificate, then click *OK*.
- For a Web server certificate, go to the Web Server page, select a different SSL mutual certificate, click *OK*, return to the Web Server page, select the correct certificate, click *OK,* then apply the changes.

44.6 When a User Accesses a Resource, the Browser Displays Certificate Errors

When you configure the Identity Server to use SSL (the HTTPS protocol), the browser must be configured to trust the CA that created the certificate for the Identity Server. If you use a well-known CA, the browser is usually already configured to trust certificates from the CA. If you use a less-known CA or the Access Manager CA to create the certificate, you need to import the public key of the trusted root certificate into the browsers to establish the trust. For the Access Manager CA, this certificate is called configCA.

For instructions on how to export the public key of a trusted root certificate, see Section 24.7, "Exporting a Public Certificate," on page 370.

To import a public key into the browser, access the certificate options, then follow the prompts:

- For Internet Explorer 7, click *Tools* > *Internet Options* > *Content* > *Certificates* > *Trusted Root Certification Authorities* > *Import*.
- For Firefox 2, click *Tools* > *Options* > *Advanced* > *Encryption* > *View Certificates* > *Authorities* > *Import*.

Appendixes

IX

The following sections contain additional documentation and information about Novell® Access Manager and the Liberty Alliance.

About Liberty

A

The Liberty Alliance is a consortium of business leaders with a vision to enable a networked world in which individuals and businesses can more easily conduct transactions while protecting the privacy and security of vital identity information.

To accomplish its vision, the Liberty Alliance established an open standard for federated network identity through open technical specifications. In essence, this open standard is a structured version of the Security Assertions Markup Language, commonly referred to as SAML, with the goal of accelerating the deployment of standards-based single sign-on technology.

For general information about the Liberty Alliance, visit the Liberty Alliance Project Web site (http://www.projectliberty.org/index.php).

Liberty resources, including specifications, white papers, FAQs, and presentations can be found at the Liberty Alliance Resources Web site (http://www.projectliberty.org/resources/index.php).

The following table provides links to specific Liberty Alliance specifications:

Table A-1 *Liberty Alliance Links*

Liberty Specification	Location
Liberty Alliance Project Overview	Liberty Alliance Project Overview (http://www.projectliberty.org/)
Liberty White Papers	Papers (http://www.projectliberty.org/liberty/resource_center/papers)
Identity Federation Specifications	Liberty ID-FF 1.2 Specification (http://www.projectliberty.org/resources/specifications.php#box1)
Web Service Framework Specifications	Liberty ID-WSF 1.1 Specifications (http://www.projectliberty.org/resources/specifications.php#box2a)
Liberty Profile Service Specifications	Liberty Alliance ID-SIS 1.0 Specifications (http://www.projectliberty.org/resources/specifications.php#box3)
Support Documentation (Glossary, Trust Model, Metadata Description, etc.)	Liberty Alliance Support Documents (http://www.projectliberty.org/resources/specifications.php#box4)
OASIS Standards (SAML)	Oasis Standards (http://www.oasis-open.org/specs/index.php#samlv2.0)

Understanding How Access Manager Uses SAML

B

Security Assertions Markup Language (SAML) is an XML-based framework for communicating security assertions (user authentication, entitlement, and attribute information) between identity providers and trusted service providers. For example, an airline company can make assertions to authenticate a user to a partner company or another enterprise application, such as a car rental company, or hotel.

The Identity Server allows SAML assertions to be exchanged with trusted service providers using SAML servers. Using SAML assertions in each Access Manager component protects confidential information by removing the need to pass user credentials between the components to handle session management.

An identity provider using the SAML protocol generates and receives assertions for authentication, according to the SAML 1.0, 1.1, and 2.0 specifications described on the Oasis Standards Web site (http://www.oasis-open.org/specs/index.php).

This section describes how Access Manager uses SAML. It includes the following topics:

- Section B.1, “Attribute Mapping with Liberty,” on page 699
- Section B.2, “Trusted Provider Reference Metadata,” on page 700
- Section B.3, “Identity Federation,” on page 700
- Section B.4, “Authorization Services,” on page 700
- Section B.5, “What's New in SAML 2.0?,” on page 700
- Section B.6, “Identity Provider Process Flow,” on page 701
- Section B.7, “SAML Service Provider Process Flow,” on page 702

B.1 Attribute Mapping with Liberty

Attribute-based authorization involves one Web site communicating identity information about a subject to another Web site in support of some transaction. However, the identity information might be some characteristic of the subject, such as a role. The attribute-based authorization is important when the subject’s identity is either not important, should not be shared, or is insufficient on its own.

In order to interoperate with trusted service providers using the SAML protocol, the Identity Server distinguishes between different attributes from different SAML implementations. All of the SAML administration is done using Liberty attributes. When the you specify which attributes to include in an assertion, or which attributes to use when locating the user from an assertion, these attributes should always be specified in the Liberty format.

In an attribute map, you convert SAML attributes from each vendor’s implementation to Liberty attributes. (See Section 7.1, “Configuring Attribute Sets,” on page 83.)

You can find detailed information about SAML 2.0 on the OASIS Security Services (SAML) TC Web site (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security).

B.2 Trusted Provider Reference Metadata

Metadata is generated by the Identity Server and is used for server communication and identification. Metadata can be obtained via URL or XML document, then entered in the system when you create the reference. Metadata is traded with federation partners and supplies various information regarding contact and organization information located at the Identity Server. Metadata is generated automatically for SAML 2.0. You enter it manually for SAML 1.1. (See Chapter 9, "Configuring Trusted Providers," on page 139.)

IMPORTANT: The SAML 2.0 and Liberty 1.2 protocols define a logout mechanism whereby the service provider sends a logout command to the trusted identity provider when a user logs out at a service provider. SAML 1.1 does not provide such a mechanism. For this reason, when a log-out occurs at the SAML 1.1 service provider, no log-out occurs at the trusted identity provider. A valid session is still running at the identity provider, and no credentials need to be entered. In order to log out at both providers, users must navigate to the identity provider that authenticated them to the SAML 1.1 service provider and log out manually.

B.3 Identity Federation

Identity federation is the association of accounts between an identity provider and a service provider, while maintaining privacy protection. From an administrative perspective, this type of sharing can help reduce identity management costs because multiple organizations do not need to independently collect and maintain identity-related data, such as passwords. From the end user's perspective, this results in an enhanced experience by requiring fewer sign-ons.

B.4 Authorization Services

When a user has authenticated to a site or application, the user has access to a resource controlled by a Policy Enforcement Point (PEP). The PEP checks for user access to the desired resource. The user is either granted or denied access to the resource. SAML is used as the communication mechanism between the PEP and a Policy Decision Point (PDP). In Novell product terminology, a PEP could be thought of as the Novell® Access Gateway, and the PDP as Novell eDirectory™ or another service.

B.5 What's New in SAML 2.0?

SAML 2.0 provides several new features:

- **Pseudonyms:** An arbitrary name assigned by the identity provider to identify a user to a service provider. The identifier has meaning only in the context of the relationship between the relying parties. They can be a principal's e-mail or account name. Pseudonyms are a key privacy-enabling feature that inhibits collusion between multiple providers.
- **Metadata:** The SAML metadata specification defines how to express configuration and trust-related data to simplify SAML deployment. Metadata identifies the Identity Servers involved in performing single sign-on between trusted identity providers and service providers.

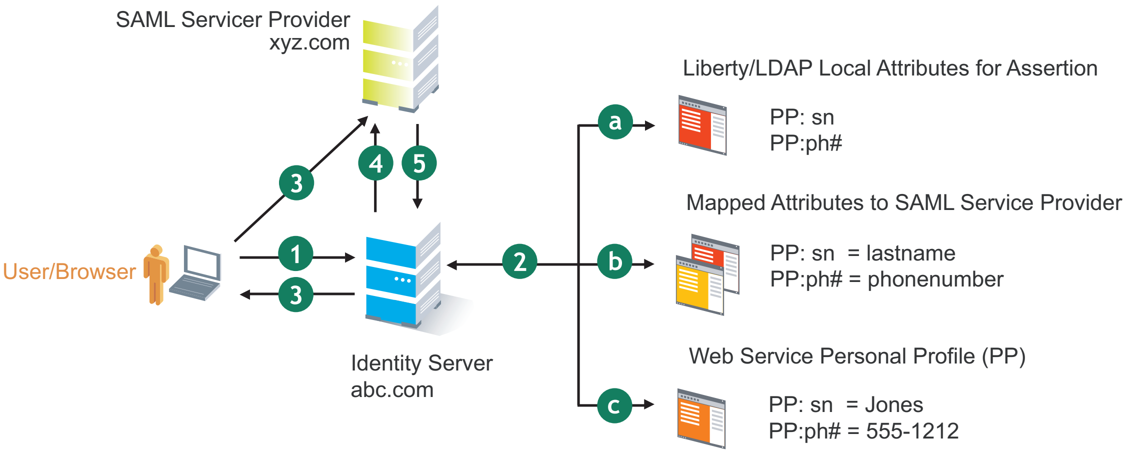
  Metadata includes supported roles, identifiers, supported profiles, URLs, certificates, and keys. System entities must agree upon the data.
- **Encryption:** SAML permits attribute statements, name identifiers, or entire assertions to be encrypted. Encryption ensures that end-to-end confidentiality of these elements can be supported as needed.

- **Attribute profiles:** These simplify how you configure and deploy systems that exchange attribute data. Attribute profiles include:
  - **Basic attribute profile:** Supports string attribute names and attribute values drawn from XML schema primitive type definitions.
  - **X.500/LDAP:** Supports canonical X.500/LDAP attribute names and values.
  - **UUID attribute profile:** Supports using UUIDs as attribute names.
  - **XACML attribute profile:** Defines formats suitable for processing by XACML (Extensible Access Control Markup Language).

B.6 Identity Provider Process Flow

The following illustration provides an example of an Identity Server automatically creating an authenticated session for the user at a trusted SAML service provider. PP indicates a Personal Profile Service as defined by the Liberty specification.

Figure B-1 *SAML Service Provider Process Flow*



1. A user is logged in to the Identity Server at abc.com (the user's identity provider) and clicks a link to xyz.com, a trusted SAML service provider.

   The Identity Server at abc.com generates the artifact. This starts the process of generating and sending the SAML assertion. An example of the HREF might be http://nidp.com/saml/genafct?TARGET=http://xyz.com/index.html&AID=XYZ.

2. The Identity Server processes attributes as follows:
   a. The server looks up LDAP or Liberty-LDAP mapped attributes. (See Section 12.9, "Mapping LDAP and Liberty Attributes," on page 184.) In this example, you use Liberty attributes such as *PP:sn* instead of *surname*. *PP:sn* and *PP:ph#* are attributes that you are sending to xyz.com.
   b. The Identity Server processes these attributes with a SAML implementation-specific attribute.

      Because the identity provider must interoperate with other SAML service providers that probably do not use consistent attribute names, you can map the service provider attributes to your Liberty and LDAP attributes on the Identity Server. In this example, the service provider names for the Liberty *PP:sn* and *PP:ph#* attributes are *lastname* and *phonenumber*, respectively. (See Section 9.8, “Selecting Attributes for a Trusted Provider,” on page 155.)

c. The Identity Server uses the PP service to look up the values for the user’s *PP:sn* and *PP:ph#* attributes.

   The Identity Server recognizes that the values for the user’s *PP:sn* and *PP:ph#* attributes are *Jones* and *555-1212*, respectively.

3. The Identity Server sends an HTTP Redirect with artifact.

   The Identity Server now has the information to generate a SAML assertion. The Identity Server sends an HTTP redirect containing the artifact back to the browser. The redirect looks something like http://xyz.com/auth/afct?TARGET=http://xyz.com/index.html&SAMLArtifact =<<artifact>>

4. The remote SAML server requests the assertion.

   The HTTP redirect results in the browser sending the artifact to the SAML server at xyz.com. The SAML server at xyz.com requests the SAML assertion from the Identity Server.

5. The Identity Server sends the assertion to the remote SAML server.

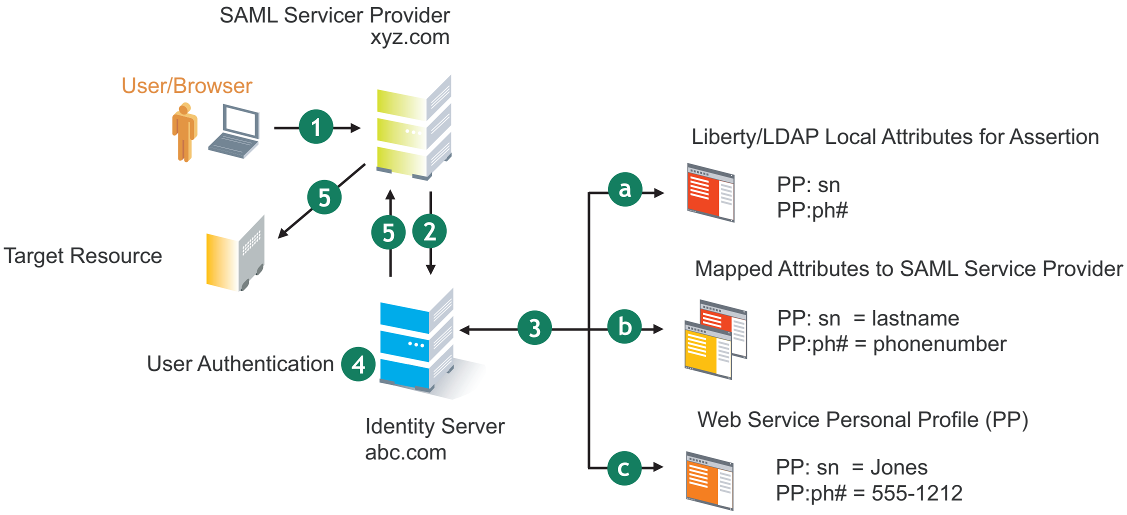
   The remote SAML server receives the artifact and looks up the assertion.The assertion is sent to the SAML server at *xyz.com* in a SOAP envelope. The assertion contains the attributes *lastname=Jones* and *phonenumber=555-1212*.

   The user now has an authenticated session at xyz.com. The xyz.com SAML server redirects the user’s browser to http://xyz.com/index.html, which was referenced in the original HREF in step 1.

B.7 SAML Service Provider Process Flow

The following illustration provides an example of the authentication process on the consumer side, when a user clicks a link at the SAML service provider (xyz.com) in order to begin an authentication session with an identity provider (such as abc.com). PP indicates a Personal Profile Service as defined by the Liberty specification.

Figure B-2 *SAML Consumer Process Flow*



1. The user clicks a link at xyz.com.

This generates a SAML assertion intended for the Identity Server at abc.com, which is the identity provider in an Access Manager configuration. After the SAML server generates the artifact, it sends the browser a redirect containing the artifact. The browser is redirected to the identity provider, which receives the artifact. The URL sent to the Identity Server looks something like: http://nidp.com/auth/afct?TARGET=http://abc.com/index.html&SAMLArtifact =<<artifact>>

2. The Identity Server at abc.com receives the assertion.

   The assertion is sent to the Identity Server packaged in a SOAP envelope. In this example, the assertion contains the attributes *lastname=Jones*, and *phonenumber=555-1212*.

3. The Identity Server determines which attributes to use when locating the user.

   The Identity Server must determine how to locate the user in the directory. When you created the SAML service provider reference for xyz.com, you specified which Liberty attributes should be used for this purpose. In this case, the you specified that *PP:sn* and *PP:ph#* should be used.

   a. The Identity Server processes the Liberty attribute map (see Section 12.9, “Mapping LDAP and Liberty Attributes,” on page 184) to the SAML implementation-specific attributes (see Section 9.8, “Selecting Attributes for a Trusted Provider,” on page 155).

      Because this SAML implementation must interoperate with other SAML implementations that probably do not use consistent attribute names, you can map the attributes used by each third-party SAML implementation to Liberty attributes on the Identity Server.

   b. The Identity Server receives implementation-specific SAML attribute names.

      The trusted service provider’s names for the Liberty *PP:sn* and *PP:ph#* attributes are returned. Using the attribute map, the Identity Server knows that the service provider’s names for these attributes are *lastname* and *phonenumber*, respectively.

   c. The Identity Server uses the PP service to lookup the values for the user’s *PP:sn* and *PP:ph#* attributes.

      The Identity Server now recognizes that the values for the user’s *PP:sn* and *PP:ph#* attributes are *Jones* and *555-1212*, respectively. The user’s DN is returned to the Identity Server, and the user is authenticated.

4. The user’s DN is returned to the Identity Server, and the user is authenticated.

5. The user is redirected to the target resource at xyz.com.

Certificates Terminology

C

A public key certificate is a collection of information attached to an electronic message. It is used to verify that the user sending the message is who he or she claims to be. The following is a list of certificate terminology used in Access Manager:

CA: A certificate authority that signs a certificate.

Certificate: Public information about the entity identified by the certificate, including the public key. A certificate is signed. The signer of the certificate (a CA), if trusted, verifies the accuracy of the information in the certificate.

Certificate Chain: In addition to identifying a user, server, or computer, certificates can validate the identity and trustworthiness of other certificates. A certificate that asserts an identity is signed by a certificate that trusts the contents of the certificate it is signing. The signing certificate in turn can be signed by another certificate, which can be signed by another certificate, and so forth, thus forming a certificate chain. The last certificate in the certificate chain is referred to as the root certificate and is a self-signed certificate.

When a certificate or certificate chain is sent from one computer to another, the receiving computer examines the certificate chain to determine if it can be trusted. To verify certificate trust in a chain, the receiving computer examines its own configuration store to see if it contains a CA certificate that matches the root certificate of the certificate chain. If so, the receiver compares its copy of the certificate with the chain's root certificate to verify its authenticity.

Certificate Signing Request (CSR): Requesting a signed certificate is accomplished by sending a CSR to the CA. A CSR is created with information about the person or organization that desires the signed certificate. A public key is also generated and included in the CSR. A private key is also generated, but not included in the CSR.

When the CA receives the CSR, the CA uses it in combination with the CA's guidelines and practices to establish that the person or organization represented by the CSR is properly identified and authorized as the owner of the information in CSR. The CA creates and signs a certificate that the requesting person or organization can use. The signature of the CA in the certificate is what identifies to anyone who trusts the CA that the entity is who it claims to be. The signed certificate is delivered to its owner, who adds it to the keystore (usually the same keystore where the private key created with the original CSR resides).

Issuer: The CA that issues a certificate.

Intermediate Certificate: A subordinate certificate issued by the trusted root specifically for end-entity server certificates. The result is a certificate chain that begins at the trusted root CA, proceeds through the intermediate certificate and ends with the SSL certificate issued to you. Using intermediate certificates adds more levels of security, but does not cause performance, installation, or compatibility issues.

Key: A certificate that also contains a private key.

Key Pair: Public and private keys generated by a cryptography system and used in combination with each other.

Keystore: A storage file containing keys, certificates, and trusted roots. Access Manager agents can access keystores to retrieve certificates, keys, and trusted roots as needed.

Local CA: The CA of the administration console's instance of eDirectory™. Also known as the Organizational CA.

Private Key: Used for authentication, data encryption/decryption, digital signing, and secure e-mail. One of the most common uses is sending and receiving digitally signed and encrypted e-mail by using the S/MIME standard.

Public Key: The publicly distributed key.

Self-Signed Certificate: A certificate whose issuer is itself.

SSL Connections: When two computers connect and need to establish trust and a secure connection, certificates are exchanged and an encryption algorithm is established. Public keys shared in the exchanged certificates, as well as the associated private keys (which are not exchanged) are used as part of the encryption algorithm. After security is established, a secure SSL session is established and the two computers are able to communicate securely.

Trusted Certificate: The certificate of a known CA. These certificates are self-signed and are recognized as representing a CA that is trusted.

Trusted Root: The same as a trusted certificate. A trusted root provides the basis for trust in public key cryptography. Trusted roots enable security for SSL, secure e-mail, and certificate-based authentication. The Identity Server already has a list of trusted certificates installed. These certificates are for root CAs, so they are called "trusted roots."

Trust Store: A keystore containing only trusted roots. Intermediate CAs and end entity public certificates can be part of a trust store.

Data Model Extension XML

D

The data model for some Web services is extensible. You can enter XML definitions of data model extensions in this field. Data model extensions hook into the existing Web service data model at predefined locations.

All schema model extensions reside inside of a schema model extension group. The group exists to bind model data items together under a single localized group name and description. Schema model extension groups can reside inside of a schema model extension root or inside of a schema model extension. There can only be one group per root or extension. Each root is hooked into the existing Web service data model. Multiple roots can be hooked into the same location in the existing Web service data model. This conceptual model applies to the structure of the XML that is required to define data model extensions.

The high-level view of the data model extension XML is as follows:

```
<SchemaExtensions>
   <Root>
     <Group>
       <Extension>
          <Group>
           <Extension>...</Extension>
           <Extension>...</Extension>
           ...
          </Group>
       </Extension>
       <Extension>
         <ValueSet>
            <Value/>
            <Value/>
         </ValueSet>
       </Extension>
      ...
    </Group>
  <Root>
<Root>...</Root>
...
</SchemaExtensions>
```

D.1 Elements

The definition of the attributes for each data model extension XML element are as follows:

- "Root Element" on page 708
- "Group Element" on page 708
- "Extension Element" on page 709
- "ValueSet Element" on page 710
- "Value Element" on page 710

Root Element

parent: The unique identifier of the “hook point” in the Web service’s data model. These hook points are defined by the Web service data model schema. These unique identifiers represent the xpaths of each data item within the model schema. Possible values for the parent attribute are listed in Table D-1:

Table D-1 *Root Element*

Personal Profile	/pp:PP/pp:Extension /pp:PP/pp:CommonName/pp:Extension /pp:PP/pp:CommonName/pp:AnalyzedName/pp:Extension /pp:PP/pp:LegalIdentity/pp:Extension /pp:PP/pp:LegalIdentity/pp:VAT/pp:Extension /pp:PP/pp:LegalIdentity/pp:AltID/pp:Extension /pp:PP/pp:EmploymentIdentity/pp:Extension /pp:PP/pp:AddressCard/pp:Extension /pp:PP/pp:AddressCard/pp:Address/pp:Extension /pp:PP/pp:MsgContact/pp:Extension /pp:PP/pp:Facade/pp:Extension /pp:PP/pp:Demographics/pp:Extension
Employee Profile	/ep:EP/ep:Extension /ep:EP/ep:CorpCommonName/ep:Extension /ep:EP/epCorpLegalIdentity/ep:Extension /ep:EP/ep:CorpLegalIdentity/ep:VAT/ep:Extension /ep:EP/ep:CorpLegalIdentity/ep:AltID/ep:Extension
Open Profile	/op:OP/op:Extension /op:OP/op:CustomizableStringsop:Extension

package (required): The Java package name where all classes for this root are implemented. This includes resource description classes and data model instance classes. For example, com.novell.nids.profile.model.extensions.

resourceClass (required): The Java class name of the resource description class that is used to load all resources associated with this root. Because resource description class files are assumed to reside in the root’s package, only the filename is needed. Resource description classes are Java classes that must be created by the person extending the model. You must also extend the com.novell.nidp.resource.NIDPResDesc class.

Group Element

resourceID: The resource ID of the display name of the group. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

descriptionResourceID: The resource ID of the description of the group. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

Extension Element

name (required): The name of the data model extension. This name must be the name of the XML element that will be used in the data model.

class (optional): The Java class name of the data model instance class. Because data model instance class files are assumed to reside in the root's package, only the filename is needed. If this attribute is omitted, then the value of the name attribute must be the instance class filename.

syntax: The syntax of this data model extension. Possible values are:

- String
- LocalizedString
- Container

format: (required if the syntax is *String* or *LocalizedString*)

The syntax of this data model extension. Possible values are:

- CaseIgnore
- CaseExtract
- URI
- URL
- Date
- DateNoYear
- CountryCode
- LanguageCode
- KeyInfo
- Number

upper: The upper bound of a numeric value. Use this attribute only if the format attribute value is Number. The value is a signed integer. If this attribute is omitted, the default value is java.lang.Integer.MAX_VALUE.

lower (optional): The lower bound of a numeric value. This attribute is only used if the format attribute value is Number. The value is a signed integer. If this attribute is omitted, the default value is java.lang.Integer.MIN_VALUE.

min (required): The cardinality of the XML element represented by this data model extension. It is the minimum number of elements allowed. The value is an unsigned integer. If this attribute is omitted, the default value is 0.

max (required): The cardinality of the XML element represented by this data model extension. It is the maximum number of elements allowed. The value is an unsigned integer. If this attribute is omitted, the default value is 1. The value UNBOUNDED may be used to indicate that there are no bounds.

namingClass: (required if syntax equals Container and max is UNBOUNDED). The class that is used as the naming attribute for the container. The class must represent one of the immediate children of the container. This class is used to name each instance of the container.

ValueSet Element

A ValueSet element contains a set of fixed values that a data model entry can contain. If a data model extension has a ValueSet, the user interface to edit the value of that extension limits the user to these values. The ValueSet element has no attributes.

Value Element

A Value element represents a value in a ValueSet. It contains the actual value to be stored in the data model entry and the display name resource ID associated with the value.

resourceID (required): The resource ID of the display name of the value. This resource ID is assumed to be a key in the resource bundle supplied by the resource description class file associated with the containing root.

value (required): The value stored in the data model entry.

name (required): The name of the data model extension. This name must be the name of the XML element that is used in the data model.

D.2 Writing Data Model Extension XML

Data model extension XML must be defined in the namespace novell:liberty:wsf:config:1:0:0 and that namespace must be defined on the SchemaExtensions element. Normally, the namespace prefix wsfc is used. An example of data model extension XML is:

```
<wsfc:SchemaExtensions xmlns:wsfc="novell:liberty:wsf:config:1:0:0">
  <wsfc:Root parent="/pp:PP/pp:Facade/pp:Extension"
     package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
     resourceClass="PPExtensionsResDesc">
   <wsfc:Group resourceId="PP.EXT.FC.GROUP"
      descriptionResourceId="PP.EXT.FC.GROUP.DESC">
   <wsfc:Extension name="AliasName"
       class="FacadeAliasName"
       syntax="String"
       format="CaseIgnore"
       resourceId="PP.EXT.FC.AliasName"
       min="0" max="1"/>
   <wsfc:Extension name="FavoriteURLs"
       class="FacadeFavoriteURLs"
       syntax="String"
       format="CaseExact"
       resourceId="PP.EXT.FC.FavoriteURLs" min="0" max="UNBOUNDED"/>
   </wsfc:Group> </wsfc:Root>
   <wsfc:Root parent="/pp:PP/pp:Demographics/pp:Extension"
      package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
      resourceClass="PPExtensionsResDesc">
   <wsfc:Group resourceId="PP.EXT.DM.GROUP"
      descriptionResourceId="PP.EXT.DM.GROUP.DESC">
   <wsfc:Extension name="EyeColor"
      class="DemographicsEyeColor"
      syntax="String" format="URI"
      resourceId="PP.EXT.DM.EyeColor"
      min="0"
      max="UNBOUNDED">
   <wsfc:ValueSet>
<wsfc:Value resourceId="PP.EXT.DM.HC.Blue" value="urn:pp:dm:blue"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Brown" value="urn:pp:dm:brown"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Green" value="urn:pp:dm:green"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Gray" value="urn:pp:dm:gray"/>
<wsfc:Value resourceId="PP.EXT.DM.HC.Hazel" value="urn:pp:dm:hazel"/>
</wsfc:ValueSet>
</wsfc:Extension>
</wsfc:Group>
</wsfc:Root>
<wsfc:Root parent="/pp:PP/pp:Extension"
   package="com.novell.nidp.liberty.wsf.idsis.ppservice.extensions"
   resourceClass="PPExtensionsResDesc">
<wsfc:Group resourceId="PP.EXT.AU.GROUP"
  descriptionResourceId="PP.EXT.AU.GROUP.DESC">
<wsfc:Extension name="Automobile"
  class="Automobile"
  syntax="Container"
  resourceId="PP.EXT.Automobile"
  min="0"
  max="UNBOUNDED"
  namingClass="AutomobileLicensePlate">
<wsfc:Group resourceId="PP.EXT.AU.DETAILS.GROUP"
  descriptionResourceId="PP.EXT.AU.DETAILS.GROUP.DESC">
<wsfc:Extension name="AutomobileModel"
  class="AutomobileModel"
  syntax="String"
  resourceId="PP.EXT.AU.Model"
  min="0"
  max="1"/>
<wsfc:Extension name="AutomobileMake"
  class="AutomobileMake"
  syntax="String"
  format="CaseIgnore"
  resourceId="PP.EXT.AU.Make"
  min="0"
  max="1"/>
<wsfc:Extension name="AutomobileLicensePlate"
  class="AutomobileLicensePlate"
  syntax="String"
  format="CaseIgnore"
  resourceId="PP.EXT.AU.LicensePlate"
  min="0" max="1"/>
</wsfc:Group>
</wsfc:Extension>
</wsfc:Group>
</wsfc:Root>
</wsfc:SchemaExtensions>
```

Logging: Using the Custom Content Filter

E

The Custom Content Filter allows you to focus trace content on a specific section of the system where you suspect a problem exists. The filter is an XML document that specifies which trace logging content to send to the trace logger.

You can limit the trace logging to one or more Java class files, or to one or more Java packages, or to one or more thread identifiers defined by Novell®. A thread identifier correlates to a group of events that logically should be logged together. In the XML, you can include and exclude content from an entity in the trace log. If trace logging content becomes too verbose, you can exclude Java classes, Java packages, or thread identifiers to reduce the irrelevant logged data.

- Section E.1, “Custom Content Filter XML Syntax,” on page 713
- Section E.2, “Examples of Custom Content Filter XML,” on page 714
- Section E.3, “Custom Content Filter Thread Identifiers,” on page 717

E.1 Custom Content Filter XML Syntax

The following text provides the XML syntax.

```
<Trace flushFrequency="immediate">
 <Classes>
  <Class>...</Class>
  <Class exclude="false">...</Class>
  <Class exclude="true">...</Class>
 </Classes>
 <Packages>
  <Package>...</Package>
  <Package exclude="false">...</Package>
  <Package exclude="true">...</Package>
 </Packages>
 <Threads>
  <ThreadId>...</ThreadId>
  <ThreadId exclude="false">...</ThreadId>
  <ThreadId exclude="true">...</ThreadId>
 </Threads>
</Trace>
```

The `<Trace>` element contains three sub-sections called `<Classes>`, `<Packages>`, and `<Threads>`. Each subsection is optional and can be omitted. The <Trace> element has a single attribute called flushFrequency that controls the frequency at which trace log data is flushed out to the file. Keep the value of this attribute set to immediate so that data is flushed as soon as possible. When in debugging mode, which is the only recommended use for trace logging, immediate flushing is preferred.

The `<Classes>` element contains zero or more `<Class>` elements. Each `<Class>` element defines a single Java class that is included or excluded in the trace log output. The name of the Java class must include the complete Java package and class name, while omitting the `.java` extension.

The `<Packages>` element contains zero or more `<Package>` elements. Each `<Package>` element defines a single Java package that is included or excluded in the trace log output. The inclusion or exclusion applied to this Java package also applies to all of this package's child packages.

The `<Threads>` element contains zero or more `<ThreadId>` elements. Each `<ThreadId>` element defines a single thread identifier defined by Novell that is included or excluded in the trace log output.

The elements `<Class>`, `<Package>`, and `<ThreadId>` have a single attribute exclude="true/false". This attribute marks the associated Java class, Java package, or Thread Identifier as being included in the trace log output or as being excluded from the trace log output. If this attribute is not present, the default is false, meaning that the default is to include the associated entity in the trace logging output.

The elements `<Class>`, `<Package>`, and `<ThreadId>` accept the single character * as the text value of the element. This wildcard character means "all entities of this type." This wildcard character can be used only as a single character. It cannot be combined with other strings in an attempt to form a wildcard string. For example `<Class>*</Class>` causes all Java classes to be included in the trace log output. However, the following example is invalid: `<Class>com.novell.nidp.NIDP*</Class>`.

E.2 Examples of Custom Content Filter XML

This section provides examples of the Custom Content Filter XML.

- Section E.2.1, "Example One," on page 714
- Section E.2.2, "Example Two," on page 715
- Section E.2.3, "Example Three," on page 716

E.2.1 Example One

The following Custom Content Filter causes all Java classes and all thread identifiers to be included in the trace log output. This filter is traces everything. Care must be taken when using this filter because large amounts of data are logged, and the performance of the system degrades substantially.

```
<Trace flushFrequency="immediate">
 <Classes>
  <Class>*</Class>
 </Classes>
 <Threads>
  <ThreadId>*</ThreadId>
 </Threads>
</Trace>
```

E.2.2 Example Two

The following Custom Content Filter causes all Java classes, except `com.novell.nidp.common.authority.ldap.jndi.JNDIUserStoreReplicaConnection`, and all thread identifiers, to be included in the trace log output. The `<Packages>` subsection is not needed because this filter already includes all Java classes. Also including all Java packages would only be redundant.

```
<Trace flushFrequency="immediate">
 <Classes>
  <Class>*</Class>
  <Class 
exclude="true">com.novell.nidp.common.authority.ldap.jndi.JNDIUse
StoreReplicaConnection</Class>
 </Classes>
 <Threads>
  <ThreadId>*</ThreadId>
 </Threads>
</Trace>
```

Specific Java classes can be excluded if irrelevant information is slowing down the log file. To determine how to filter out unwanted entries from the trace log content, perform the following steps:

1. Locate the header for the entries that you want to exclude from the trace log.

   Each trace entry in the log file has a header that names the Java class where the trace entry originated. An example header is:

   ```
   NIDP TRACE LOG Method:
   com.novell.nidp.liberty.wsf.WSFFramework.initialize().
   ```
2. Extract the Java class or Java package name from the header.

   In the above example, the Java class is

   `com.novell.nidp.liberty.wsf.WSFFramework`

   and the Java package is `com.novell.nidp.liberty.wsf`. The `.initialize()` method is inside the WSFFramework class. You do not need to extract the method name. You can ignore it.
3. Add a `<Class>` or `<Package>` entry to the Custom Content Filter XML that excludes the Java class or Java package.

   Excluding the entire package removes trace log entries from other Java class files in the same package. Using the preceding example, if you want to exclude trace log entries from only the `Java class com.novell.nidp.liberty.wsf.WSFFramework` entry you would add

   ```
   <Class 
   exclude="true">com.novell.nidp.liberty.wsf.WSFFramework</Class>
   ```

   to the `<Classes>` element subsection. If you want to exclude the entire package, you add

   ```
   <Package exclude="true">com.novell.nidp.liberty.wsf</Package>
   ```

   to the `<Packages>` element subsection.

   If you follow the preceding steps to exclude a Java class or package, and the trace log entry is still logged, this is because the log entry is being logged based on a thread identifier. Logs based on thread identifiers do not consider the Java class or package when deciding if the trace log should occur. In this case, determine which aspect of the product the trace log entry pertains to, and attempt to match it with a thread identifier. (Thread identifiers are explained in Section E.2.3, “Example Three,” on page 716.) Then add a

```
<ThreadId exclude="true">[thread id name]</ThreadId>
```

line to the `<Threads>` subsection. Or, if you want to remove all trace logs associated with all thread identifiers, simply remove the `<Threads>` subsection.

E.2.3 Example Three

The following Custom Content Filter example includes all packages except for the explicitly excluded `com.novell.nidp.common.authority.ldap` package.

```
<Trace flushFrequency="immediate">
 <Packages>
  <Package>*</Package>
  <Package exclude="true">com.novell.nidp.common.authority.ldap</
Package>
 </Packages>
 <Threads>
  <ThreadId>*</ThreadId>
  <ThreadId exclude="true">tIdPPModelTokenCreate</ThreadId>
  <ThreadId exclude="true">tIdEPModelTokenCreate</ThreadId>
  <ThreadId exclude="true">tIdCUPModelTokenCreate</ThreadId>
  <ThreadId exclude="true">tIdAPModelTokenCreate</ThreadId>
  <ThreadId exclude="true">tIdCPModelTokenCreate</ThreadId>
  <ThreadId exclude="true">tIdWSFSchemaExtensions</ThreadId>
  <ThreadId exclude="true">tIdRequestResponse</ThreadId>
  <ThreadId exclude="true">tIdConfiguration</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiConnShare</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiOperations</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiOperationStats</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiSearch</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiGetObject</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiModifyObject</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiCreateConnection</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiCloseConnection</ThreadId>
  <ThreadId exclude="true">tIdLdapJndiPerUserStoreStats</ThreadId>
  <ThreadId exclude="true">tIdCBPing</ThreadId>
  <ThreadId exclude="true">tIdCBRetirement</ThreadId>
  <ThreadId exclude="true">tIdCBLogouts</ThreadId>
  <ThreadId exclude="true">tIdCBStats</ThreadId>
  <ThreadId exclude="true">tIdHealthCheck</ThreadId>
 </Threads>
</Trace>
```

This example shows the filter for one of the most verbose packages. The example shows that you have chosen to exclude the LDAP package because the issue under investigation was not related to LDAP. This example goes on to include all thread identifiers, and then excludes each thread identifier. Thus, all thread identifiers are excluded by this filter. However, this example shows the complete list of thread identifiers. Therefore, using this filter would require you to only change `exclude="true"` to `exclude="false"` in order to include the relevant thread identifier.

E.3 Custom Content Filter Thread Identifiers

A thread identifier names a sequence of events that can be traced as a group. The events logged for a given thread identifier can be a sequence of events performed to accomplish a task, or it can be a group of similar events.

The Web Service Framework includes several Web services. Each Web service has a data model associated with it. As the identity provider or service provider initializes, the data model builds the set of data items included in each Web service. Including all of the following five thread identifiers logs the compete data model of the indicated service. This log is written once at startup and once each time the identity server application is restarted.

- **tIdPPModelTokenCreate:** Logs the data model of the Personal Profile Service.
- **tIdEPModelTokenCreate:** Logs the data model of the Employee Profile Service.
- **tIdCUPModelTokenCreate:** Logs the data model of the Custom Profile Service.
- **tIdAPModelTokenCreate:** Logs the data model of the Role Profile Service.
- **tIdCPModelTokenCreate:** Logs the data model of the Credential Profile Service.

As each Web Service data model is built, you can configure model extensions (or schema extensions) to add additional data items to the model. You can configure the model extensions for each Web Service by adding XML to the edit box on each Web Service's Details: General Settings page (*Identity Servers > Servers > Edit > Liberty > Web Service Provider > [Profile] > Details*).

The following thread identifier logs each new entry that is added to the model. Also, all errors that occur from attempting to add to the model are logged.

- **tIdWSFSchemaExtensions:** Logs successful and failed additions to all Web service data models.

One of the best ways to debug the identity provider or service provider is to log the HTTP requests and HTTP responses that are handled by the identity provider or service provider. The following thread identifier logs the requests and responses for all subsystems. The HealthCheck request is not logged under this thread identifier because it might become verbose and regularly interferes with locating pertinent data. Therefore, the HealthCheck request is only logged if the tIdHealthCheck thread identifier is included.

- **tIdRequestResponse:** Logs the requests and responses for all subsystems.

As the identity provider or service provider is initializing after a startup or a reconfigure, the configuration is applied to the identity provider or service provider. The following thread identifier logs the configuration data that is used to initialize the identity provider or service provider.

- **tIdConfiguration:** Logs the versions of various subcomponents used in the system. Also logs the details of each Web service.

The identity provider or service provider include an LDAP operations subsystem that handles all communications with the LDAP trust/configuration database and LDAP user stores. This subsystem maintains connection pools for general purpose administrative level LDAP operations and for user LDAP operations. A typical administrative LDAP operation is to read a user's identity information from the directory. A typical user LDAP operation is to bind a user to a directory object to prove that a name/password combination is valid.

As the system is pushed to its limits, the LDAP operations subsystem can determine that it needs more connections devoted to administrator operations. Thus, user connections from the user connection pool are shared with the administrator connection pool. This also can happen in the opposite direction. The following thread identifiers log data about the current state of the LDAP operations subsystem and the LDAP operations it performs. The LDAP operations subsystem is the most verbose logging section of the identity provider or service provider. Thus, there is a different thread identifier for each basic LDAP operation. Be careful when including all of these thread identifiers at the same time because large amounts of data are logged.

- **tIdLdapJndiConnShare:** Logs details about how the LDAP operations subsystem shares the connection between user and administrator connection pools.
- **tIdLdapJndiOperations:** Logs details associated with the LDAP operations subsystem.
- **tIdLdapJndiOperationStats:** Periodically logs the LDAP operations subsystem statistics.
- **tIdLdapJndiSearch:** Logs details about all LDAP Object Search operations.
- **tIdLdapJndiGetObject:** Logs details about all LDAP Object Get operations.
- **tIdLdapJndiModifyObject:** Logs details about all LDAP Object Modify operations.
- **tIdLdapJndiCreateConnection:** Logs details about all LDAP Connection Create operations.
- **tIdLdapJndiCloseConnection:** Logs details about all LDAP Connection Close operations.
- **tIdLdapJndiPerUserStoreStats:** Periodically logs generic statistics about each user store.

The Session Broker is a component of the embedded service provider that works closely with the Access Gateway to monitor user authentications within a clustered environment. As users log in to the system, their login information is registered in the Session Broker of the embedded service provider. The Session Broker communicates with other members of the cluster to share user session information. Therefore, successful communication between cluster members is vital to a properly functioning system.

The session broker is also responsible for timing-out or retiring authentication data that has been unused for too long. When an authentication data item times out, or when the user logs out of the system, the session broker is responsible to send a message to each Access Gateway in the cluster to tell the Access Gateway that the logout has taken place, and that user’s authentication data must be removed.

- **tIdCBPing:** Logs a periodic ping that displays all cluster members to which successful communication is available. The computer initiating the ping is not shown in the list.
- **tIdCBRetirement:** Logs details about user session data that is being retired.
- **tIdCBLogouts:** Logs details about the messages sent to the Access Gateway indicating that a user session has timed out or was logged out.
- **tIdCBStats:** Logs generic statistics about the Session Broker.

A periodic health check of the system can be configured. The following thread identifier logs the details about the system items checked during the health check. If health reports an error and the administrator is not sure why the error is happening, then this health check log detail can provide more information.

- **tIdHealthCheck:** Logs details about the health check.

Authentication Classes and Duplicate Common Names

F

If users have the same common name and exist in different containers under the same authentication search base, one or more attributes in addition to the common name must be configured for authentication to uniquely identify the user. You can set up an authentication class to handle duplicate common names.

1. Select either the name/password or secure name/password class.
2. Add two properties to the class:
   - **Query:** The value of the Query attribute needs to be a valid LDAP query string. Field names from the JSP login form can be used in the LDAP query string as variables for LDAP attribute values. The variables must be enclosed between two % characters. For example, (&(objectclass=person)(cn=%Ecom_User_ID%)(mail=%Ecom_Email%)) queries for an object of type person that contained a common name equal to the Ecom_User_ID field from the specified JSP form and mail equal to the Ecom_Email field from the same JSP form.
   - **JSP:** The JSP property value needs to be the name of a new `.jsp` file that includes all the needed fields for the Query property. The value of this attribute does not include the `.jsp` extension of the file. For example, if you create a new `.jsp` file named `login2.jsp`, the value of the JSP property is login2.

Access Manager Audit Events and Data

G

The sections contains all the Novell® audit events logged by Access Manager. Each event has the EventID, Description, Originator Title, Target Title, Subtarget Title, Text1 Title, Text2 Title, Text3 Title, Value1 Title, Value1 Type, Group Title, Data Length, and Data Type values stored. Each field contains a single character token (such as B, U, Y, and so on) that represent the data fields of the audit event, with each letter representing a different data field. The mapping of the character tokens to data fields is found in the `nids_en.lsc` and `sslvpn_en.lsc` files.

Novell Access Manager is listed among the log applications on the *General* tab on the Logging Server Options page (*Auditing and Logging* > *Logging Server Options*). You can view events on the Event list page in *Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*.

When you run an SQL query (*Auditing and Logging* > *Queries* > *[Name]* > *Run*), the system displays the results on the Query Results page. The *EventID* column displays the description of the event. Below, the event ID is listed with the description, to help you quickly locate the data for each audit event. For instructions on how to set up Novell Audit to use a SQL database and generate queries, see "Creating Novell Audit Queries" in the *Novell Access Manager 3.0 SP3 IR2 Setup Guide*.

This section discusses the following audit events:

G.1 NIDS: Sent a Federate Request (002e0001)

This event is generated when you select the *Federation Request Sent* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Sent a federate request.

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.2 NIDS: Received a Federate Request (002e0002)

This event is generated when you select the *Federation Request Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Received a federate request.

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier; Data Description: Service Provider ID

Text2 (T): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.3 NIDS: Sent a Defederate Request (002e0003)

This event is generated when you select the *Defederation Request Sent* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Sent a defederate request.

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier; Data Description: Service Provider ID

Text2 (T): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.4 NIDS: Received a Defederate Request (002e0004)

This event is generated when you select the *Defederation Request Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Received a defederate request

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier
Data Description: Service Provider ID

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.5 NIDS: Sent a Register Name Request (002e0005)

Description: NIDS: Sent a register name request

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.6 NIDS: Received a Register Name Request (002e0006)

This event is generated when you select the *Register Name Request Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Received a register name request

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.7 NIDS: Logged Out an Authentication that Was Provided to a Remote Consumer (002e0007)

This event is generated when you select the *Logout Provided* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Logged out an authentication that was provided to a remote consumer

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Timed Out
Data Description: 0 = other reason
1 = timed out

Group (G): 0

Data Length (X): 0

Data (D): null

G.8 NIDS: Logged out a Local Authentication (002e0008)

This event is generated when you select the *Logout Local* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Logged out a local authentication

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Timed Out
Data Description: 0 = other reason
1 = timed out

Group (G): 0

Data Length (X): 0

Data (D): null

G.9 NIDS: Provided an Authentication to a Remote Consumer (002e0009)

This event is generated when you select the *Login Consumed* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Provided an authentication to a remote consumer

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: Authentication Type
Data Description: Authentication Profile

Text2 (T): Schema Title: Authentication Entity Name
Data Description: Authentication Source

Text3 (F): Schema Title: Contract Class or Method Name
Data Description: Authentication Contract URI

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.10 NIDS: User Session Was Authenticated (002e000a)

This event is generated when you select the *Login Provided* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: User session was authenticated

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text1 (S): Schema Title: Authentication Type
Data Description: Authentication Profile

Text2 (T): Schema Title: Authentication Entity Name
Data Description: Authentication Source

Text3 (F): Schema Title: Contract Class or Method Name
Data Description: Authentication Contract URI

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.11 NIDS: Failed to Provide an Authentication to a Remote Consumer (002e000b)

This event is generated when you select the *Login Consumed Failure* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to provide an authentication to a remote consumer

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Provider Identifier
Data Description: Service Provider ID

Text3 (F): Schema Title: Reason
Data Description: Reason Message

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.12 NIDS: User Session Authentication Failed (002e000c)

This event is generated when you select the *Login Provided Failure* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration. Use the *Description* field and the *Text3 (F)* field to determine whether the failure came from a contract, SAML 1.1, SAML 2.0, or Liberty.

Description: `NIDS: User session authentication failed`. This string plus one of the following phrases: for a contract failure, `Contract Execution`; for a SAML 1.1 failure, `SAML Assertion`; for a SAML 2.0 failure, `SAML2 SSO`; for a Liberty failure, `Liberty SSO`.

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Authentication Contract Name
Data Description: Contract URI

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Reason
Data Description: Reason Message

Text3 (F): Schema Title: Authentication Source
Data Description: For a contract, contains the authentication method name; for Liberty, contains the service provider IP; for SAML 1.1, contains the SAML assertion issuer; for SAML 2.0, contains the service provider IP.

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.13 NIDS: Received an Attribute Query Request (002e000d)

This event is generated when you select the *Attribute Query Request Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Received an attribute query request

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: LDAP Auth: User DN
Other Auth: User GUID

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier
Data Description: Service Provider ID

Text2 (T): Schema Title: Attribute Names
Data Description: Requested Attributes

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.14 NIDS: User Account Provisioned (002e000e)

This event is generated when you select the *User Account Provisioned* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: User account provisioned

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Identifier
Data Description: Displayable user name

SubTarget (Y): null

Text1 (S): Schema Title: User Identifier
Data Description: Authentication User Name

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.15 NIDS: Failed to Provision a User Account (002e000f)

This event is generated when you select the *User Account Provisioned Failure* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Failed to provision a user account

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Identifier
Data Description: Displayable User Name

SubTarget (Y): null

Text1 (S): Schema Title: User Identifier
Data Description: Authentication User Name

Text2 (T): Schema Title: Reason
Data Description: Reason Message

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.16 NIDS: Web Service Query (002e0010)

This event is generated when you select the *Web Service Query Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration. The Identity Server uses this event for two types of Web service queries:

- **Discovery:** This is a query to discover a service. For this type of query, the *Group (G)* field is not used. For a remote query, the *Data Description* of the *Value1* field is set to 0. For a local query, the *Data Description* of the *Value1* field is set to 1.
- **Profile:** This is a query to get attributes for a user from a profile (personal, credential, etc.). For this type of query, the *Group (G)* field contains a GroupingID for all attributes selected in the request. A separate event is generated for each attribute select list in the request. For a remote query, the *Data Description* of the *Value1* field is set to 0. For a local query, the *Data Description* of the *Value1* field is set to 1.

Description: NIDS: Web Service query

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier
Data Description: Requesting Provider ID

Text2 (T): Schema Title: Select String
Data Description: Requested attributes; select string

Text3 (F): Schema Title: Service Identifier
Data Description: Web Service URI

Value1 (1): Schema Title: Local
Data Description: 0 – Remote
1 – Local

Group (G): Schema Title: Query Group
Data Description: If this is a profile query, contains the grouping ID for all attributes selected in this request. Otherwise, it is not used in the event.

Data Length (X): 0

Data (D): null

G.17 NIDS: Web Service Modify (002e0011)

This event is generated when you select the *Web Service Modify Handled* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration. The Identity Server uses this event for two types of Web service modify requests:

- **Discovery:** This is a request to discover a service to modify. For this type of request, the *Group (G)* field is not used. For a remote request, the *Data Description* of the *Value1* field is set to 0. For a local request, the *Data Description* of the *Value1* field is set to 1.

- **Profile:** This is a request to modify the attributes of a user in a profile (personal, credential, etc.). For this type of request, the *Group (G)* field contains a GroupingID for all attributes selected in the request. A separate event is generated for each attribute select list in the modify request. For a remote request, the *Data Description* of the *Value1* field is set to 0. For a local request, the *Data Description* of the *Value1* field is set to 1.

Description: NIDS: Web Service modify

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Provider Identifier
Data Description: Requesting Provider ID

Text2 (T): Schema Title: Select String
Data Description: Modified attributes select string

Text3 (F): Schema Title: Service Identifier
Data Description: Web Service URI

Value1 (1): Schema Title: Local
Data Description: 0 – Remote; 1 – Local

Group (G): Schema Title: Modify Group
Data Description: If this is a profile modify, contains the grouping ID for each attribute select list in the request. Otherwise, it is not used in the event.

Data Length (X): 0

Data (D): null

G.18 NIDS: Connection to User Store Replica Lost (002e0012)

This event is generated when you select the *LDAP Connection Lost* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Connection to user store replica lost

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Replica Name
Data Description: Replica name

SubTarget (Y): null

Text1 (S): Schema Title: User Store Replica Host
Data Description: IP Address of User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.19 NIDS: Connection to User Store Replica Reestablished (002e0013)

This event is generated when you select the *LDAP Connection Reestablished* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Connection to user store replica reestablished

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Store Replica Name

Data Description: Replica name

SubTarget (Y): null

Text1 (S): Schema Title: User Store Replica Host
Data Description: IP Address of User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.20 NIDS: Server Started (002e0014)

This event is generated when you select the *Server Started* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Server started

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier
Data Description: Configuration Object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier
Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.21 NIDS: Server Stopped (002e0015)

This event is generated when you select the *Server Stopped* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Server stopped

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier
Data Description: Configuration object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier
Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.22 NIDS: Server Refreshed (002e0016)

This event is generated when you select the *Server Refreshed* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Server Refreshed

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Configuration Identifier
Data Description: Configuration Object DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier
Data Description: Unique server ID also used to create Liberty and SAML artifacts

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.23 NIDS: Intruder Lockout (002e0017)

This event is generated when you select the *Intruder Lockout Detected* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Intruder Lockout

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Server Identifier
Data Description: IP Address of User Store replica server

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.24 NIDS: Severe Component Log Entry (002e0018)

This event is generated when you select the *Component Log Severe Messages* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Severe Component Log Entry

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Component Log Text
Data Description: Server Error Text

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.25 NIDS: Warning Component Log Entry (002e0019)

This event is generated when you select the *Component Log Warning Messages* option under *Novell Audit Logging* on the Logging page of an Identity Server configuration.

Description: NIDS: Warning Component Log Entry

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Component Log Text
Data Description: Warning Error Text

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.26 NIDS: Roles PEP Configured (002e0300)

This event is generated for Identity Server roles.

Description: NIDS: Roles PEP Configured

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): Schema Title: Policy Enforcement List Length
Data Description: byte length of PEL

Data (D): Schema Title: Policy Enforcement List
Data Description: Policy Enforcement List (PEL) data

G.27 Access Gateway: PEP Configured (002e0301)

This event is generated when you enable auditing.

Description: Access Gateway: PEP configured

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Audit Enabled
Data Description: 0 = No; 1 = Yes

Group (G): 0

Data Length (X): Schema Title: Policy Enforcement List Length
Data Description: byte length of PEL

Data (D): Schema Title: Policy Enforcement List
Data Description: Policy Enforcement List (PEL) data

G.28 J2EE Agent: Web Service Authorization PEP Configured (002e0305)

This event is generated when you enable auditing.

Description: 2EE Agent: Web Service Authorization PEP Configured

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: Audit Enabled
Data Description: 0 = Yes; 1 = No

Group (G):

Data Length (X): Schema Title: Protected Resource List Length
Data Description: byte length of PWRL

Data (D): Schema Title: Protected Resource List
Data Description: Protected Web Resource List (PWRL)

G.29 J2EE Agent: JACC Authorization PEP Configured (002e0306)

This event is generated when you enable auditing.

Description: J2EE Agent: JACC Authorization PEP Configured

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text2 (T): null

Text3 (F): null

Value1 (1): Schema Title: audit enabled
Data Description: 0 = No; 1 = Yes

Group (G):

Data Length (X): Schema Title: Protected Resource List Length
Data Description: byte length of PWML

Data (D): Schema Title: Protected Resource List
Data Description: Protected Web Module List (PWML)

G.30 Roles Assignment Policy Evaluation (002e0320)

This event is generated when you enable auditing.

Description: Roles assignment policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Assigned Roles
Data Description: Assigned Role or error message

Text3 (F): Schema Title: Policy Action
Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.31 Access Gateway: Authorization Policy Evaluation (002e0321)

This event is generated when you enable auditing.

Description: Access Gateway: Authorization policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action
Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.32 Access Gateway: Form Fill Policy Evaluation (002e0322)

This event is generated when you enable auditing.

Description: Access Gateway: Form Fill policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action
Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.33 Access Gateway: Identity Injection Policy Evaluation (002e0323)

This event is generated when you enable auditing.

Description: Access Gateway: Identity Injection policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action
Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.34 J2EE Agent: Web Service Authorization Policy Evaluation (002e0324)

This event is generated when you enable auditing.

Description: J2EE Agent: Web Service Authorization policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource URL
Data Description: Protected resource URL

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): Schema Title: Policy Action
Data Description: Policy Action FDN

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.35 J2EE Agent: Web Service SSL Required Policy Evaluation (002e0325)

This event is generated when you enable auditing.

Description: J2EE Agent: Web Service SSL Required policy evaluation

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): null

Value1 (1): Schema Title: SSL Required
Data Description: 0 = No; 1 = Yes

Group (G): 0

Data Length (X): 0

Data (D): null

G.36 J2EE Agent: Startup (002e0401)

This event is generated when you select the *Startup, shutdown, and reconfigure* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Startup

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.37 J2EE Agent: Shutdown (002e0402)

This event is generated when you select the *Startup, shutdown, and reconfigure* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Shutdown

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.38 J2EE Agent: Reconfigure (002e0403)

This event is generated when you select the *Startup, shutdown, and reconfigure* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Reconfigure

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.39 J2EE Agent: Authentication Successful (002e0404)

This event is generated when you select the *Successful authentications* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Authentication successful

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.40 J2EE Agent: Authentication Failed (002e0405)

This event is generated when you select the *Unsuccessful authentications* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Authentication failed

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.41 J2EE Agent: Web Resource Access Allowed (002e0406)

This event is generated when you select the *Allowed web resource access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Web Resource access allowed

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Source IP Address
Data Description: User IP Address

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: Web resource permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.42 J2EE Agent: Clear Text Access Allowed (002e0407)

This event is generated when you select the *Allowed clear text access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Clear text access allowed

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Source IP Address
Data Description: User IP Address

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: Web User Data Permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.43 J2EE Agent: Clear Text Access Denied (002e0408)

This event is generated when you select the *Denied clear text access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Clear text access denied

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Source IP Address
Data Description: User IP Address

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: Web User Data Permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.44 J2EE Agent: Web Resource Access Denied (002e0409)

This event is generated when you select the *Denied web resource access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: Web resource access denied

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): Schema Title: Source IP Address
Data Description: User IP Address

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: Web User Data Permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.45 J2EE Agent: EJB Access Allowed (002e040a)

This event is generated when you select the *Allowed EJB access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: EJB access allowed

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: EJB Method Permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.46 J2EE Agent: EJB Access Denied (002e040b)

This event is generated when you select the *Denied EJB access* option in the *Audit Configuration* section of the Server Configuration page for the J2EE Agents.

Description: J2EE Agent: EJB access denied

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: User Identifier
Data Description: User DN

SubTarget (Y): null

Text1 (S): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text2 (T): Schema Title: Permission Requested
Data Description: EJB Method Permission

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.47 Access Gateway: Access Allowed (0x002e0504)

This event is generated when you select the *Access Allowed* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Access Allowed.

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0504

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name
Data Description: Configured Name of Protected Resource

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address
Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

G.48 Access Gateway: Access Denied (0x002e0505)

This event is generated when you select the *Access Denied* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Access Denied

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0505

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name
Data Description: Configured Name of Protected Resource

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address
Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

G.49 Access Gateway: URL Not Found (0x002e0508)

This event is generated when you select the *URL Not Found* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: URL Not Found

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0508

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address
Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

G.50 Access Gateway: System Started (0x002e0509)

This event is generated when you select the *System Started* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: System Started

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0509

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.51 Access Gateway: System Shutdown (0x002e050a)

This event is generated when you select the *System Shutdown* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: System Shutdown

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e050a

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): null

Text1 (S): null

Text2 (T): null

Text3 (F): null

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.52 Access Gateway: Identity Injection Parameters (0x002e050c)

This event is generated when you select the *Identity Injection Parameters* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Identity Injection Parameters

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e050c

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Injection Location
Data Description: 2710 – Auth Header 2720 – Custom Header
2730 – Query Parameters

Group (G): 0

Data Length (X): 0

Data (D): null

G.53 Access Gateway: Identity Injection Failed (0x002e050d)

This event is generated when you select the *Identity Injection Failed* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Identity Injection Failed

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e050d

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Injection Location
Data Description: 2710 – Auth Header 2720 – Custom Header
2730 – Query Parameters

Group (G): 0

Data Length (X): 0

Data (D): null

G.54 Access Gateway: Form Fill Authentication (0x002e050e)

This event is generated when you select the *Form Fill Success* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Form Fill Authentication

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e050e

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name
Data Description: Configured name of protected resource

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.55 Access Gateway: Form Fill Authentication Failed (0x002e050f)

This event is generated when you select the *Form Fill Failed* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: Form Fill Authentication Failed

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e050f

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): Schema Title: Protected Resource Name
Data Description: Configured name of protected resource

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.56 Access Gateway: URL Accessed (0x002e0512)

This event is generated when you select the *URL Accessed* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: URL Accessed

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0512

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address
Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

G.57 Access Gateway: IP Access Attempted (0x002e0513)

This event is generated when you select the *IP Access Attempted* option on the Novell Audit page of an Access Gateway.

Description: Access Gateway: IP Access Attempted

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0513

Originator (B): Schema Title: Originator
Data Description: JCC Device ID (AMDEVICEID#device_id:)

Target (U): null

SubTarget (Y): Schema Title: Protected Resource URL
Data Description: Protected Resource URL

Text1 (S): Schema Title: User Identifier
Data Description: User DN

Text2 (T): Schema Title: Authentication Identifier
Data Description: IDP Session ID (AMAUTHID#auth_id:)

Text3 (F): Schema Title: Event Identifier
Data Description: Event Tracking Identifier

Value1 (1): Schema Title: Source IP Address
Data Description: User IP address (numeric format – host order)

Group (G): 0

Data Length (X): 0

Data (D): null

G.58 Management Communication Channel: Health Change (0x002e0601)

This event is generated when you select the *Health Changes* option on the Access Manager Auditing page.

Description: Management Communication Channel: Health Change

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0601

Originator (B): Schema Title: Originator
Data Description: "devmanagement" (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Changed Device
Data Description: IP address and device type of changed device

Text2 (T): Schema Title: Old State
Data Description: Old State

Text3 (F): Schema Title: New State
Data Description: New State

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.59 Management Communication Channel: Device Imported (0x002e0602)

This event is generated when you select the *Server Imports* option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Imported

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0602

Originator (B): Schema Title: Originator
Data Description: "devmanagement" (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device
Data Description: IP address and device type of changed device

Text2 (T): blank string

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.60 Management Communication Channel: Device Deleted (0x002e0603)

This event is generated when you select the *Server Deletes* option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Deleted

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0603

Originator (B): Schema Title: Originator
Data Description: "devmanagement" (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device
Data Description: IP address and device type of changed device

Text2 (T): Schema Title: Administrator
Data Description: DN of administrator deleting the device

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.61 Management Communication Channel: Device Configuration Changed (0x002e0604)

This event is generated when you select the *Configuration Changes* option on the Access Manager Auditing page.

Description: Management Communication Channel: Device Configuration Changed

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0604

Originator (B): Schema Title: Originator
Data Description: "devmanagement" (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device
Data Description: IP address and device type of changed device

Text2 (T): Schema Title: Administrator
Data Description: DN of administrator invoking the configuration change

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null

G.62 Management Communication Channel: Device Alert (0x002e0605)

This event is generated when you enable auditing.

Description: Management Communication Channel: Device Alert

In the Event list (*Auditing and Logging* > *Logging Server Options* > *[Name of Novell Audit Secure Logging Server]* > *Novell Access Manager* > *Events*), this column is called *Event Name*.

In a query, this column is called *EventID*.

Event ID: 0x002e0605

Originator (B): Schema Title: Originator
Data Description: "devmanagement" (AMDEVICEID#devmanagement:)

Target (U): null

SubTarget (Y): null

Text1 (S): Schema Title: Device
Data Description: IP address of device generating the alert

Text2 (T): Schema Title: Alert Message
Data Description: alert message string

Text3 (F): blank string

Value1 (1): 0

Group (G): 0

Data Length (X): 0

Data (D): null