NetIQ® Certificate Server™ 8.8 SP8

Administration Guide

September 2013

Legal Notice

Contents

About this Book and the Library

NetIQ Certificate Server provides public key cryptography services that are natively integrated into eDirectory and that allow you to mint, issue, and manage both user and server certificates. These services allow you to protect confidential data transmissions over public communications channels such as the Internet.

The *Administration Guide* describes the functionality of NetIQ Certificate Server, how to set it up, and how to manage it. This book also provides some basic information about how public key cryptography works.

For the most recent version of the *NetIQ Certificate Server 8.8 SP8 Administration Guide*, see the NetIQ eDirectory 8.8 online documentation (https://www.netiq.com/documentation/edir88/) Web site.

Intended Audience

The guide is intended for network administrators.

This guide is available at the NetIQ eDirectory 8.8 documentation Web site (https://www.netiq.com/documentation/edir88/).

For information about the eDirectory management utility, see the *NetIQ iManager 2.7.7 Administration Guide* (https://www.netiq.com/documentation/imanager/imanager_admin/data/bookinfo.html).

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. If you have suggestions for improvements, click **Add Comment** at the bottom of any page in the HTML versions of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

Qmunity, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, Qmunity helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

1 Overview

NetIQ Certificate Server 8.8.8 is part of eDirectory 8.8 SP8. For eDirectory 8.8 and later, Certificate Server is automatically installed when you install eDirectory. For information about eDirectory 8.8 SP8, including supported platforms and installation instructions, see the *NetIQ eDirectory 8.8 SP8 Administration Guide* (https://www.netiq.com/documentation/edir88/edir88/data/bookinfo.html).

Certificate Server provides public key cryptography services that are natively integrated into eDirectory and that allow you to mint, issue, and manage both user and server certificates. These services allow you to protect confidential data transmissions over public communications channels such as the Internet.

NOTE: If you are unfamiliar with public key cryptography concepts, see "Public Key Cryptography Basics" on page 85.

1.1 NetIQ Certificate Server Features

Public key cryptography presents unique challenges to network administrators. NetIQ Certificate Server helps you meet these challenges in the following ways:

- Provides public key cryptography services on your network

  You can create an Organizational Certificate Authority (CA) within your eDirectory tree, allowing you to issue an unlimited number of user and server certificates. You can also use the services of an external certificate authority, or use a combination of both as your needs dictate.
- Controls the costs associated with obtaining and managing public key certificates

  You can create an Organizational CA and issue public key certificates through the Organizational CA.
- Allows public key certificates to be openly available while also protecting them against tampering

  Certificates are stored in eDirectory and can therefore leverage eDirectory replication and access control features.
- Allows private keys to be accessible to only the software routines that use them for signing and decrypting operations

  Private keys are encrypted by Novell International Cryptography Infrastructure (NICI) and made available only to the software routines using them for signing and decrypting operations.
- Securely backs up private keys

  Private keys are encrypted by NICI, stored in eDirectory, and backed up by using standard eDirectory backup utilities.
- Allows central administration of certificates using iManager.

iManager plug-ins are provided, allowing you to manage certificates issued from your Organizational CA or from any other CA that supports a certificate signing request in PKCS #10 format.

- Allows users to manage their own certificates

  Users can use iManager to export keys for use in cryptography-enabled applications without system administrator intervention.

- Supports popular e-mail clients and browsers

  NetIQ Certificate Server allows you to create and manage user certificates for securing e-mail. NetIQ Certificate Server supports GroupWise 5.5 or later, Microsoft Outlook 98 and Outlook 2000, Netscape Messenger, and other popular e-mail clients. It's also compatible with Mozilla, Mozilla Firefox, and Microsoft Internet Explorer browsers.

1.2 NetIQ Certificate Server Components

This section describes the components of NetIQ Certificate Server.

1.2.1 NetIQ Certificate Server

NetIQ Certificate Server consists of the PKI server component and a plug-in module to iManager. iManager is the administration points for NetIQ Certificate Server.

NetIQ Certificate Server allows you to request, manage, and store public key certificates and their associated key pairs in the eDirectory tree, and to establish an Organizational Certificate Authority that is specific to your eDirectory tree and your organization.

NetIQ Certificate Server derives all supported cryptography and signature algorithms, as well as supported key sizes, from Novell International Cryptographic Infrastructure (NICI). Therefore, a single version of NetIQ Certificate Server can be used in installations throughout the world.

After installing NetIQ Certificate Server, you manage it by using iManager.

You can use iManager to perform the following tasks:

Use 4096 Bit Keys in Certificates

eDirectory 8.8 supports key sizes up to 4096 bits. However, in order to use key sizes larger than 2048 bits, you must upgrade all of the servers in your eDirectory Tree to eDirectory 8.8 and upgrade all clients to NICI 2.7.0 or later. For more information on upgrading NICI, see the *Novell International Cryptographic Infrastructure 2.7 Administration Guide* (https://www.netiq.com/documentation/nici27x/nici_admin_guide/data/agbe6d0.html). Also, if you plan to use 4 KB certificates with your applications, the applications must support 4 KB keys or they do not work properly.

Note that 4 KB keys take significantly more time to generate and use.

Certificate Server lets you select the key size as part of any certificate creation procedure.

Create an Organizational Certificate Authority for Your Organization

During the installation, you can elect to create an Organizational Certificate Authority (CA) if one does not already exist in the eDirectory tree. You can also create or re-create the Organizational CA after the installation is completed.

The Organizational CA object contains the public key, private key, certificate, certificate chain, and other configuration information for the Organizational CA. The Organizational CA object resides in the Security container in eDirectory.

After a server is configured to provide the certificate authority service, it performs that service for the entire eDirectory tree.

For more information on creating an Organizational CA, see Section 2.2, "Creating an Organizational Certificate Authority Object," on page 20.

Create a Server Certificate Object for Each Cryptography-Enabled Application

The Certificate Server installation creates default Server Certificate objects.

- SSL CertificateDNS - *server_name*
- A certificate for each IP address configured on the server (IPAG*xxx.xxx.xxx.xxx - server_name*)
- A certificate for each DNS name configured on the server (DNSAG*www.example.com - server_name*)

NOTE: eDirectory 8.8 SP8 does not automatically create SSL CertificateIP. SSL CertificateDNS contains all the IPs listed in the Subject Alternative Name.

You can create other Server Certificate objects after the installation is completed.

The Server Certificate object contains the public key, private key, certificate, and certificate chain that enables SSL security services for server applications. Server Certificate objects can be signed by either the Organizational CA or by an external CA.

A server can have many Server Certificate objects associated with it. Any cryptography-enabled applications running on a particular server can be configured to use any one of the Server Certificate objects for that server. Multiple applications running on a given server can use the same Server Certificate object; however, a Server Certificate object cannot be shared between servers.

You can create Server Certificate objects only in the container where the server resides. If the Server object is moved, all Server Certificate objects belonging to that server must be moved as well. You should not rename a Server Certificate object. You can determine which Server Certificate objects belong to a server by searching for the server's name in the Server Certificate Object Name or by looking at the host server field when viewing the Server Certificate object in iManager.

The key pair stored in the Server Certificate object is referenced by the name you enter when the key pair is created. The key pair name is not the name of the Server Certificate object. When configuring cryptography-enabled applications to use key pairs, you reference those keys by their key pair name, not by the Server Certificate object name.

If the default Server Certificate objects become corrupted or invalid, use the Create Default Certificates Wizard to replace the old default certificates. For information on how to access the Create Default Certificates Wizard, see Section 3.2.2, "Creating Default Server Certificate Objects," on page 37.

Create a User Certificate

Users have access to their own user certificates and private keys, which can be 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 using the S/MIME standard.

Generally, only the CA administrator has sufficient rights to create user certificates. However, only the user has rights to export or download the private key from eDirectory. Any user can export any other user's public key certificate.

The user certificate is created from the *Security* tab of the user's property page and is signed by the Organizational CA. Certificates and private keys created by other CAs can be imported after being created.

Multiple certificates can be stored on the user's object.

For more information on creating a user certificate, see Section 2.6.1, "Creating a User Certificate," on page 24.

Create a Trusted Root Container

A trusted root provides the basis for trust in public key cryptography. Trusted roots are used to validate certificates signed by other CAs. Trusted roots enable security for SSL, secure e-mail, and certificate-based authentication.

A Trusted Root Container is an eDirectory object that contains Trusted Root objects.

The default Trusted Root Container is CN=trusted roots.CN=security.

For more information on creating a Trusted Root Container, see Section 2.6.2, "Creating a Trusted Root Container," on page 25.

Create a Trusted Root Object

A Trusted Root object is an eDirectory object that contains a CA's Trusted Root certificate that is known to be authentic and valid. The Trusted Root Certificate can be exported and used as needed. Applications that are configured to use the Trusted Root Certificate consider a certificate valid if it has been signed by one of the CAs in the Trusted Root Container.

The Trusted Root object must reside in a Trusted Root Container.

For more information on creating a Trusted Root object see Section 2.6.3, “Creating a Trusted Root Object,” on page 25.

Create Certificates For External Users and Servers

The CA administrator can use the Organizational CA to sign certificates for users and servers outside of eDirectory. Such certificates are requested using a PKCS#10 Certificate Signing Request (CSR) provided to the CA administrator in an out-of-band fashion.

Given a CSR, the CA administrator can issue the certificate by using the Issue Certificate tool in iManager. The resulting certificate is not stored in an object in eDirectory. It must be returned to the requestor in an out-of-band fashion.

Validate Certificates

NetIQ Certificate Server allows you to check the validity of any certificate in the eDirectory tree. The certificate validation process checks each certificate in the certificate chain back to the trusted root certificate and returns a status of Valid or Invalid.

- To check the validity of certificates for the Organizational CA, see Section 3.1.10, “Validating the Organizational CA's Certificates,” on page 35.
- To check the validity of certificates for a server, see Section 3.2.12, “Validating a Server Certificate,” on page 44.
- To check the validity of certificates for a user, see Section 3.3.7, “Validating a User Certificate,” on page 49.
- To check the validity of certificates for a Trusted Root, see Section 3.6.5, “Validating a Trusted Root Object,” on page 58.

Certificates are considered valid if they pass a predefined set of criteria including whether the current time is within the validity period of the certificate, whether it has not been revoked, and whether it has been signed by a CA that is trusted.

When validating user certificates or intermediate CA certificates in CN=trusted roots.CN=security signed by external CAs, the external CA’s certificate must be stored in a Trusted Root object in order for the certificate validation to be successful.

Manage Certificate Revocation Lists

A Certificate Revocation List (CRL) is a published list of revoked certificates and the reason the certificates were revoked.

NetIQ Certificate Server provides a system for managing CRLs. This is an optional system, but it must be implemented if you want to be able to revoke certificates created by the Organizational CA. For more information on managing CRLs, see Section 3.7, “Certificate Revocation List (CRL) Tasks,” on page 59.

During the Certificate Server installation, a CRL container is created if the user has the appropriate rights to create it. If not, the CRL container can be created manually by someone with the appropriate rights after the installation is completed.

A CRL Configuration object can be created in the CRL container. The object contains the configuration information for the CRL objects that are available in the eDirectory tree. Normally, you have only one CRL Configuration object in your tree. You might need multiple CRL Configuration objects if you are creating or rolling over a new Organizational CA, but only one CRL Configuration object can be used to create new certificates.

A CRL object, also known as a distribution point, can be created in any container in the eDirectory tree. However, NetIQ CRL objects usually reside in a CRL container. A CRL object is automatically created for you when you create a CRL Configuration object. The CRL object contains a CRL file, which contains the detailed CRL information. For a NetIQ CRL object, the CRL file is automatically created and updated whenever the server issues a new one. For other CRL objects, you must import a CRL file from a third-party CA.

Deleting a CRL Configuration object is possible, but it is not recommended. When a CRL Configuration object is deleted, the server quits creating the CRL files. If a CRL file already exists in the location specified in the CRL object, certificate validation continues to use it until it expires. After it expires, all certificates that have a CRL distribution point that references that CRL file fail validation.

If you delete a CRL object, it is re-created the next time the server generates the CRL file. If you delete a CRL object that you created using iManager and import it, then it is gone permanently and any certificates that reference it are considered invalid.

The general rule is to not delete a CRL container, CRL Configuration object, CRL object, or CRL file until one issue date after the last certificate that contains a related distribution point has expired.

Export Private Keys and Certificates

User, server, and CA keys can be marked as exportable when they are created. 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 (PFX or PKCS #12), which allows it to be transported to other platforms. It is encrypted with a user-specified password to protect the private key.

Exporting private keys and certificates can be done to obtain a backup copy of the key, to move the key to a different server, or to share the key between servers.

For more information on exporting private keys and certificates, see Section 3.3.6, “Exporting a User Certificate and Private Key,” on page 48.

Import Private Keys and Certificates

You can choose to import a key rather than create a new one at the time a server certificate, a user certificate, or a CA object is created. The key and its associated certificates must be in PFX or PKCS #12 format.

You might choose to import a key rather than create a new one for a CA object to recover from a server failure, to move the Organizational CA from one server to another, or for a CA that is subordinate to another CA.

You might choose to import a user certificate or private key if it has been signed by a third-party CA.

You might choose to import a key rather than create a new one for a Server Certificate object to recover from a server failure, to move the key and certificate to another server, or to share the key and certificate with another server.

Create an SAS Service Object

The SAS service object facilitates communication between a server and its server certificates. If you remove a server from an eDirectory tree, you also need to delete the SAS service object associated with that server. If you want to put the server back into the tree, you must create the SAS service object to go with that server. If you do not, you cannot create new server certificates.

The SAS service object is automatically created as part of the server health check. You should not need to create it manually.

You can create a new SAS service object only if there is not a properly named SAS service object in the same container as the server object. For example, for a server named WAKE, you will have a SAS service object named SAS Service - WAKE. The utility adds the DS pointers from the Server object to the SAS object, and from the SAS object to the Server object, as well as set up the correct ACL entries on the SAS service object.

If a SAS service object already exists with the proper name, you cannot create a new one. The old SAS service object's DS pointers might be wrong or missing, or the ACLs might not be correct. In this case, you can delete the corrupt SAS service object and use iManager to create a new one. If there are server certificates that belong to this server, you need to link them to the SAS service object manually by using the *Other* tab.

For more information on creating a SAS service object, see Section 2.6.4, "Creating an SAS Service Object," on page 26.

1.2.2 Novell International Cryptographic Infrastructure

Novell International Cryptographic Infrastructure (NICI) is the underlying cryptographic infrastructure that provides the cryptography for NetIQ Certificate Server, NetIQ Modular Authentication Services (NMAS™), and other applications.

NICI must be installed on the server in order for NetIQ Certificate Server to work properly. NICI does not ship with NetIQ Certificate Server. In most cases NICI is provided and installed when NetIQ Certificate Server is bundled with another product, such as Open Enterprise Server (OES) or eDirectory. If you need a newer version of NICI, you can download it from the Novell Downloads Web site (http://www.novell.com/download).

1.2.3 For Additional Information

For instructions on installing NetIQ Certificate Server when it is included with other NetIQ or Novell products, see the installation guide for that product.

For instructions on setting up NetIQ Certificate Server, see Chapter 2, "Setting Up NetIQ Certificate Server," on page 19.

For information about administering NetIQ Certificate Server, see Chapter 3, "Managing NetIQ Certificate Server," on page 27.

For the latest online documentation for this and other NetIQ products, see the NetIQ Documentation Web site (https://www.netiq.com/documentation/).

For additional information about this and other security products and technologies, see the NetIQ Web site (https://www.netiq.com/products/).

2 Setting Up NetIQ Certificate Server

After you install NetIQ Certificate Server, you must set it up for use on your network by completing the following tasks:

2.1 Deciding Which Type of Certificate Authority to Use

NetIQ Certificate Server allows you to create certificates for both servers and end users. Server certificates can be signed by either the Organizational CA or by an external or third-party CA. User certificates can be signed only by the Organizational CA; however, you can import user certificates signed by a third-party CA in PKCS#12 format.

During the Server Certificate object creation process, you are asked which type of certificate authority will sign the Server Certificate object.

The Organizational Certificate Authority is specific to your organization and uses an organizational-specific public key for signing operations. The private key is created when you create the Organizational Certificate Authority.

A third-party certificate authority is managed by a third party outside of the eDirectory tree. An example of a third party certificate authority is VeriSign.

Both types of certificate authorities can be used simultaneously. Using one type of certificate authority does not preclude the use of the other.

2.1.1 Benefits of Using an Organizational Certificate Authority Provided with NetIQ Certificate Server

- **Compatibility.** The Organizational Certificate Authority is compatible with NetIQ or Novell applications such as LDAP services, Portal Services, and the Apache Web Server. Certificates issued by the Organizational CA are X.509 v3 compliant and can also be used by third-party applications.

- **certificate authorityCost savings.** The Organizational Certificate Authority lets you create an unlimited number of public key certificates at no cost; obtaining a single public key certificate through an external Certificate Authority might cost a significant amount of money.
- **Component of a complete and compatible solution.** By using the Organizational Certificate Authority, you can use the complete cryptographic system built into eDirectory without relying on any external services. In addition, NetIQ Certificate Server is compatible with a wide range of NetIQ or Novell products.
- **Certificate attribute and content control.** An Organizational Certificate Authority is managed by the network administrator, who decides on public key certificate attributes such as certificate life span, key size, and signature algorithm.
- **Simplified management.** The Organizational Certificate Authority performs a function similar to external certificate authorities but without the added cost and complexity.

2.1.2 Benefits of Using an External Certificate Authority

- **Liability.** An external certificate authority might offer some liability protection if, through the fault of the certificate authority, your private key was exposed or your public key certificate was misrepresented.
- **Availability.** An external certificate authority's certificate might be more widely available and more widely trusted by applications outside of eDirectory.

2.2 Creating an Organizational Certificate Authority Object

By default, the NetIQ Certificate Server installation process creates the Organizational Certificate Authority (CA) for you. You are prompted to specify an Organizational CA name. When you click *Finish*, the Organizational CA is created with the default parameters and placed in the Security container.

If you want more control over the creation of the Organizational CA, you can create the Organizational CA manually by using iManager. Also, if you delete the Organizational CA, you need to re-create it.

During the creation process, you are prompted to name the Organizational Certificate Authority object and to choose a server to host the Organizational CA service (the server the Organizational CA service will run on). In determining the server to host the Organizational CA service, consider the following:

- Select a server that is physically secure.

  Physical access to the CA server is an important part of the security of the system. If the CA server is compromised, all certificates issued by the CA are also compromised.
- Select a server that is highly available, stable, and robust.

  If the CA service is not available, certificates cannot be created. This affects installation of new servers because certificates need to be created during install.
- Select a server that only runs software you trust.

  Running unknown or questionable software might compromise the CA service.
- Select a server that will not be removed from the tree.

  If the server is removed from the tree, you need to either re-create the CA object by using a backup you made before removing the CA, or you need to create a new CA. If you create a new CA, you might need to replace your existing server and user certificates.

- Select a server that runs a protocol that is compatible with other servers in your tree.

  Examples are IP, IPX, or IP/IPX.

To create the Organizational Certificate Authority object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

   If no Organizational Certificate Authority object exists, this opens the Create an Organizational Certificate Authority Object dialog box and the corresponding wizard that creates the object. Follow the prompts to create the object. For specific information on the dialog box or any of the wizard pages, click Help.
4. After you have finished creating the Certificate Authority, we recommend that you make a backup of the CA's public/private key pair and store this in a safe and secure place. See "Backing Up an Organizational CA" on page 32.

NOTE: You can have only one Organizational CA for your eDirectory tree.

2.3 Subordinate Certificate Authority

NetIQ Certificate Server has added support for a Subordinate Certificate Authority. This feature allows the Organizational CA to be subordinate to either a third-party CA or to a CA in another eDirectory tree. You still can have only one Organizational CA in your eDirectory tree.

The following are some of the reasons to have a Subordinate CA:

- Allows the Organizational CA to become part of an existing third-party PKI
- Allows multiple trees to share a common PKI Trusted Root (or Trust Anchor)
- Allows for greater security of the Root CA by having the CA reside on a more secure system
- Provides less risk by having the Root CA reside in a tree that is more tightly managed (for example, in a tree protected from rogue-administrators/users)

2.3.1 Creating a Subordinate Certificate Authority

In order to create a Subordinate CA, you must first delete the existing Organizational CA (see "Deleting the Organizational CA" on page 35). You must already have a PKCS#12 file containing the public/private keys and the certificate chain for the Subordinate CA. You can either obtain this file directly from a third-party CA or use Section 2.3.2, "Creating a PKCS#12 File for a Subordinate CA," on page 22 to learn how to create one. In order to create the Subordinate CA, connect to the tree in iManager and use the Configure Certificate Authority task, using the Import creation method.

2.3.2 Creating a PKCS#12 File for a Subordinate CA

1. Create a Server Certificate object (or KMO) and a PKCS#10 CSR.
   - **1a** Launch iManager.
   - **1b** On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create Server Certificate*.
   - **1c** Select the server that will eventually host the CA, specify a certificate nickname, select the Custom creation method, then click *Next*.
   - **1d** Select *External Certificate Authority*, then click *Next*.
   - **1e** Select a key size (2048 bit is recommended), make sure that *Allow Private Key to Be Exported* is selected, then click *Next*.
   - **1f** Click the *Edit* button to the right of the *Subject name* field and edit the *Subject name* to reflect the subordinate CA and tree, select the Signature algorithm (SHA-1 is currently recommended), then click *Next*.
   - **1g** Verify that the summary is correct, then click *Finish*.
   - **1h** Click *Save Certificate Signing Request*, then follow the prompts to save the CSR to a file.
2. Get the CSR signed to create a certificate.
   - **2a** If the Subordinate CA is to be part of a third-party PKI, have the third-party CA create the certificate from the CSR.

     or

     If the Subordinate CA is to be signed by a CA in another eDirectory tree, continue with Step 2b.
   - **2b** Launch iManager.
   - **2c** On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Issue Certificate*.
   - **2d** Select the file containing the CSR, then click *Next*.
   - **2e** Select a key type of *Certificate Authority*, deselect *Enable Extended Key Usage*, then click *Next*.
   - **2f** Select the Certificate Authority Certificate type, select either the *Unspecified* or a *Specific Path* length, then click *Next*.
   - **2g** Verify the subject name and edit it if necessary. Specify a validity period (5-10 years is recommended), then click *Next*.
   - **2h** Select a format for the certificate, then click *Next*.
   - **2i** Click *Finish*.
   - **2j** Click *Download the issued certificate*, then follow the prompts to save the certificate.
3. Acquire the CA certificates.
   - **3a** If the Subordinate CA is to be part of a third-party PKI, acquire the CA certificates from the third party.

     or

     If the Subordinate CA is to signed by a CA in another eDirectory tree, continue with Step 3b.
   - **3b** Launch iManager.
   - **3c** On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.
   - **3d** Click the *Certificates* tab, then select *Self-Signed Certificate*.
   - **3e** Click *Export*.

3f Do not export the private key and select a format for the certificate, then click *Next*.

3g Click *Save the Exported Certificate to a File*, then follow the prompts to save the certificate.

4 Import the certificates into the Server Certificate object (or KMO).

4a Launch iManager.

4b On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.

4c Select the Server Certificate object (or KMO) created in Step 1, then click *OK*.

4d Click the *Certificates* tab, then click *Import*.

4e Select the two files containing the certificates acquired in Step 2 and Step 3, then click *OK*.

5 Export the public/private keys to a PKCS#12 file.

5a Continuing from Step 4e, click *Export*, choose to include the private key, then click *Next*.

5b Click *Save the Exported Certificate to a File*, then follow the prompts to save the PKCS#12 file.

5c Make a copy of this file and store it in a secure place along with the password.

6 (Optional) Delete the Server Certificate object (or KMO).

2.4 Creating a Server Certificate Object

Server Certificate objects are created in the container that holds the server's eDirectory object. Depending on your needs, you might create a separate Server Certificate object for each cryptography-enabled application on the server, or you might create one Server Certificate object for all applications used on that server.

NOTE: The terms Server Certificate object and Key Material object (KMO) are synonymous. The schema name of the eDirectory object is NDSPKI:Key Material.

When you install NetIQ Certificate Server, the Server Certificate object is created with the default parameters and placed in the container where the target server resides. If you ever need to overwrite or create new default certificates, you can use the Create Default Certificates Wizard. See Section 3.2.2, “Creating Default Server Certificate Objects,” on page 37.

If you want more control over the creation of the Server Certificate object, you can create the Server Certificate object manually. You can also create additional Server Certificate objects.

2.4.1 Manually Creating a Server Certificate Object

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create Server Certificate*.

This opens the Create a Server Certificate dialog box and the corresponding wizard that creates the Server Certificate object. Follow the prompts to create the object. For specific information on the dialog box or any of the wizard pages, click *Help*.

2.4.2 Hints for Creating Server Certificates

During the Server Certificate object creation process, you are prompted to name the key pair and choose the server that the key pair will be associated with. The Server Certificate object is generated by NetIQ Certificate Server, and its name is based on the key pair name that you choose.

If you choose the Custom creation method, you are also prompted to specify whether the Server Certificate object will be signed by your organization's Organizational Certificate Authority or by an external certificate authority. For information about making this decision, see Section 2.1, "Deciding Which Type of Certificate Authority to Use," on page 19.

If you decide to use your organization's Organizational CA, the server that the Server Certificate object is associated with must be able to communicate with the server that hosts the Organizational CA, or it must be the same server. These servers must be running the same protocol (IP/IPX).

If you decide to use an external certificate authority to sign the certificate, the server that the Server Certificate object is associated with generates a certificate signing request that you need to submit to the external certificate authority.

After the certificate is signed and returned to you, you need to install it into the Server Certificate object, along with the trusted root for the external Certificate Authority.

After you have created the Server Certificate object, you can configure your applications to use it. (See Section 2.5, "Configuring Cryptography-Enabled Applications," on page 24.) Keys are referenced in the application's configuration by the key pair name that you entered when you created the Server Certificate object.

2.5 Configuring Cryptography-Enabled Applications

After you have configured NetIQ Certificate Server, you must configure your individual cryptography-enabled applications so that they can use the custom certificates that you created. The configuration procedures are unique to the individual applications, so we recommend that you consult the application's documentation for specific instructions.

See Section 3.9, "Application Tasks," on page 66 for specific instructions on configuring GroupWise® 5.5 or later, Outlook* 98, Outlook 2000, and Netscape Messenger.

2.6 Additional Components to Set Up

NetIQ Certificate Server includes some additional components that can be set up to provide additional functionality.

2.6.1 Creating a User Certificate

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create User Certificate*.

This opens a wizard that helps you create the user certificate. Follow the prompts to create the object. For specific information on the wizard pages, click *Help*.

2.6.2 Creating a Trusted Root Container

You can create a Trusted Root container anywhere in the eDirectory tree.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create Trusted Root Container*.

4 Specify a name for the Trusted Root container.

5 Browse and select the context for the Trusted Root container.

6 Click *OK*.

NOTE: Different applications might require that the Trusted Root container be given a specific name and be in a specific location in the eDirectory tree. NetIQ Certificate Server requires that the Trusted Root container be named Trusted Roots and be located in the Security container. The certificates in this container are used to validate user certificates signed by external CAs and intermediate CA certificates stored in Trusted Root objects. Server certificates and the Organizational CA's certificates use the certificate chain stored in their own objects.

2.6.3 Creating a Trusted Root Object

A Trusted Root object can only reside in a Trusted Root container.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create Trusted Root*.

This opens the Create a Trusted Root Object Wizard that helps you create the Trusted Root object. Follow the prompts to create the object. For specific information on the wizard pages, click *Help*.

NOTE: Any type of certificate can be stored in a Trusted Root object (CA certificates, intermediate CA certificates, or user certificates).

2.6.4 Creating an SAS Service Object

The SAS Service object is automatically created as part of the server health check. You should not need to create it manually. If you need to create it manually, use the following procedure:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create SAS Service Object*.

   This opens the Create a SAS Service Object Wizard that helps you create the SAS Service object. Follow the prompts to create the object. For specific information on the wizard pages, click *Help*.

3 Managing NetIQ Certificate Server

As a system administrator, you need to perform several tasks to maintain the public key cryptography services provided through NetIQ Certificate Server. Use iManager to perform these tasks. This section provides a brief overview and specific information on completing each task.

Certificate Authority tasks:

Server Certificate object tasks:

- Section 3.7.10, “Viewing a CRL Object's Properties,” on page 64
- Section 3.7.11, “Deleting a CRL Object,” on page 65

eDirectory tasks:

- Section 3.8.1, “Resolving Multiple Security Containers, Organizational CAs, KAP Containers, and W0 Objects,” on page 65
- Section 3.8.2, “Restoring or Re-creating a Security Container,” on page 66
- Section 3.8.3, “Restoring or Re-creating KAP and W0,” on page 66

Application tasks:

- Section 3.9.1, “Importing the User Certificate and Private Key into Your E-Mail Client,” on page 67
- Section 3.9.2, “Configuring Your E-Mail Client to Secure Your E-Mail,” on page 68
- Section 3.9.3, “Configuring Your Browser or E-Mail Client to Accept Certificates,” on page 70
- Section 3.9.4, “Configuring Microsoft Internet Explorer (IE) for SSL with NetIQ Certificates,” on page 72
- Section 3.9.5, “Configuring Microsoft IIS for Client Authentication with NetIQ Certificates,” on page 72
- Section 3.9.6, “Requesting a Server Certificate for Microsoft IIS,” on page 73

3.1 Certificate Authority Tasks

- Section 3.1.1, “Creating an Organizational Certificate Authority Object,” on page 29
- Section 3.1.2, “Issuing a Public Key Certificate,” on page 30
- Section 3.1.3, “Viewing the Organizational CA's Properties,” on page 30
- Section 3.1.4, “Viewing an Organizational CA's Public Key Certificate Properties,” on page 30
- Section 3.1.5, “Viewing the CA's Self-Signed Certificate Properties,” on page 31
- Section 3.1.6, “Exporting the Organizational CA's Self-Signed Certificate,” on page 31
- Section 3.1.7, “Backing Up an Organizational CA,” on page 32
- Section 3.1.8, “Restoring an Organizational CA,” on page 33
- Section 3.1.9, “Moving the Organizational CA to a Different Server,” on page 34
- Section 3.1.10, “Validating the Organizational CA's Certificates,” on page 35
- Section 3.1.11, “Deleting the Organizational CA,” on page 35
- Section 3.1.12, “Rolling Over an Organizational CA,” on page 36

3.1.1 Creating an Organizational Certificate Authority Object

This task is described in Section 2.2, “Creating an Organizational Certificate Authority Object,” on page 20.

3.1.2 Issuing a Public Key Certificate

This task allows you to generate certificates for cryptography-enabled applications that do not recognize Server Certificate objects.

Your Organizational CA works the same way as an external CA. That is, it has the ability to issue certificates from certificate signing requests (CSRs). You can issue certificates using your Organizational CA when a user sends a CSR to you for signing. The user requesting the certificate can then take the issued certificate and import it directly into the cryptography-enabled application.

To issue a public key certificate:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Issue Certificate*.
4. Use the *Browse* button to locate a CSR file, open the file, then click *Next*.
5. Specify the key type, the key usage, and the extended key usage, then click *Next*.
6. Specify the certificate basic constraints, then click *Next*.
7. Specify the subject name, the validity period, the effective and expiration dates, and any custom extensions, then click *Next*.
8. Review the parameters sheet. If it is correct, click *Finish*. If not, click *Back* until you reach the point where you need to make changes.

   When you click *Finish*, a dialog box explains that a certificate has been created. You can save the certificate to the system clipboard in Base64 format, to a Base64-formatted file, or to a binary DER-formatted file. You can also click *Details* to view details about the issued certificate.

3.1.3 Viewing the Organizational CA's Properties

In addition to the eDirectory rights and properties that can be viewed with any eDirectory object, you can also view properties specific to the Organizational CA, including the properties of the public key certificate and the self-signed certificate associated with it.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

   This brings up the property pages for the Organizational CA, which include a General page, a CRL page, and a Certificates page.
4. Click the tabs you want to view.

3.1.4 Viewing an Organizational CA's Public Key Certificate Properties

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

   This brings up the property pages for the Organizational CA, which include a General page, a CRL page, a Certificates page, and other eDirectory-related pages.

4 Click *Certificates*, then click the nickname of the public key certificate you want to view.

5 To view the certificate chain, click the plus sign (+) in front of the certificate's nickname to expand the view.

6 Click *Close*.

3.1.5 Viewing the CA's Self-Signed Certificate Properties

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

   This brings up the property pages for the Organizational CA, which include a General page, a CRL page, and a Certificates page.

4 Click *Certificates*, then click the nickname of the self-signed certificate you want to view.

   If this is a Subordinate CA, there is no self-signed certificate.

5 To view the certificate chain, click the plus sign (+) in front of the certificate's nickname to expand the view.

6 Click *Close*.

3.1.6 Exporting the Organizational CA's Self-Signed Certificate

The self-signed certificate can be used for verifying the identity of the Organizational CA and the validity of a certificate signed by the Organizational CA.

From the Organizational CA's property page, you can view the certificates and properties associated with this object. From the self-signed certificate property page, you can export the self-signed certificate to a file for use in cryptography-enabled applications.

The self-signed certificate that resides in the Organizational CA is the same as the Trusted Root certificate in a Server Certificate object that has a certificate signed by the Organizational CA. Any service that recognizes the Organizational CA's self-signed certificate as a trusted root can accept a valid user or server certificate signed by the Organizational CA.

This task does not apply if the CA is a Subordinate CA.

To export the Organizational CA's self-signed certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

   This brings up the property pages for the Organizational CA, which include a General page, a CRL page, a Certificates page, and other eDirectory-related pages.

4 Click *Certificates*, then select the self-signed certificate.

5 Click *Export* and follow the prompts to export the certificate.

   This starts the Certificate Export Wizard. Ensure the *Export private key* check box is not selected (does not have a check mark).

6 Click *Finish*.

3.1.7 Backing Up an Organizational CA

NetIQ recommends that you back up your Organizational CA's private key and certificates in case the Organizational CA's host server has an unrecoverable failure. If a failure should occur, you can use the backup file to restore your Organizational CA to any server in the tree that has Certificate Server version 2.21 or later installed.

NOTE: The ability to back up an Organizational CA is available only for Organizational CAs created with Certificate Server version 2.21 or later. In previous versions of Certificate Server, the Organizational CA's private key was created in a way that made exporting it impossible.

The backup file contains the CA's private key, self-signed certificate, public key certificate, and several other certificates necessary for it to operate. This information is stored in PKCS #12 format (also known as PFX).

The Organizational CA should be backed up when it is working properly.

With Certificate Server 3.2 and later, in order to completely back up the Certificate Authority, it is necessary to back up the CRL database and the Issued Certificates database.

For other platforms, both of these databases are located in the same directory as the eDirectory `dib` files. The defaults for these locations are as follows:

- Windows: `c:\novell\nds\dibfiles`
- Linux/AIX/Solaris: `/var/opt/novell/edirectory/data/dib`

These defaults can be changed at the time that eDirectory is installed.

The files to back up for the CRL database are `crl.db`, `crl.01` and the `crl.rfl` directory. The files to back up for the Issue Certificates database are `cert.db`, `cert.lck`, `cert.01`, and the `cert.rfl` directory.

The eDirectory `dib` directory should be part of a standard and regular backup plan.

To back up the Organizational CA:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

4 Click *Certificates*, then select either the *Self Signed Certificate* or the *Public Key Certificate*. Both certificates are written to the file during the backup operation.

5 Click *Export*.

   This opens a wizard that helps you export the certificates to a file.

6 Choose to export the private key, specify a password with 6 or more alphanumeric characters to use in encrypting the PFX file, then click *Next*.

7 Click the *Save the exported certificate to a file* link and provide the filename and the location for the backup file.

8 Click *Save*.

9 Click *Close*.

The encrypted backup file is written to the location specified. It is now ready to be stored in a secure location for emergency use.

IMPORTANT: The exported file should be put on a diskette or some other form of backup media and stored in a secure place. The password used to encrypt the file should be committed to memory or stored in a safe place to ensure that it is available when needed, but inaccessible to others.

3.1.8 Restoring an Organizational CA

If the Organizational CA object has been deleted or corrupted, or if the Organizational CA's host server has suffered an unrecoverable failure, the Organizational CA can be restored to full operation through using a backup file created as described in Section 3.1.7, "Backing Up an Organizational CA," on page 32.

The ability to restore an Organizational CA is only available in Certificate Server version 2.21 or later.

NOTE: If you were unable to make a backup of the Organizational CA, the Organizational CA might still be recovered if NICI 2.*x* is installed on the server and a backup was made of the NICI configuration information.

With Certificate Server 3.2 and later, in order to completely restore the Certificate Authority, it is necessary to restore the CRL database and the Issued Certificates database.

For other platforms, both of these databases are located in the same directory as the eDirectory `dib` files. The defaults for these locations are as follows:

- Windows: `c:\novell\nds\dibfiles`
- Linux/AIX/Solaris: `/var/opt/novell/edirectory/data/dib`

These defaults can be changed at the time that eDirectory is installed.

The files to restore for the CRL database are `crl.db`, `crl.01` and the `crl.rfl` directory. The files to restore for the Issue Certificates database are `cert.db`, `cert.lck`, `cert.01`, and the `cert.rfl` directory.

The eDirectory `dib` directory should be part of a standard and regular backup plan.

To restore the Organizational CA:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 (Conditional) If the Organizational CA object still exists, you need to delete it:

3a On the *Roles and Tasks* menu, click *Directory Administration* > *Delete Object*.

3b Browse to and click the Organizational CA object.

3c Click *OK*.

4 From the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.

This opens the Create an Organizational Certificate Authority Object dialog box and the corresponding wizard that creates the object

5. In the creation dialog box, specify the server that should host the Organizational CA and the name of the Organizational CA object. The server specified must have Certificate Server version 2.21 or higher installed and be up and running.
6. Select the *Import* option.
7. Click *Next*.
8. Click *Read from File*, then select the name of the backup file in the dialog box.
9. Click *Open*.
10. Enter the password used to encrypt the file when the backup was made.
11. Click *Finish*.

    The Organizational CA's private key and certificates have now been restored and the CA is fully functional. The backup file can now be stored again for future use.

IMPORTANT: Be sure to protect your backup media.

3.1.9 Moving the Organizational CA to a Different Server

You can move your Organizational CA from one server to another by using the backup and restore procedures outlined in "Backing Up an Organizational CA" on page 32 and "Restoring an Organizational CA" on page 33.

With Certificate Server 3.2 and later, in order to completely move the Certificate Authority, it is necessary to move the CRL database and the Issued Certificates database.

For other platforms, both of these databases are located in the same directory as the eDirectory `dib` files. The defaults for these locations are as follows:

- Windows: `c:\novell\nds\dibfiles`
- Linux/AIX/Solaris: `/var/opt/novell/edirectory/data/dib`

These defaults can be changed at the time that eDirectory is installed.

The files to move for the CRL database are `crl.db`, `crl.01` and the `crl.rfl` directory. The files to move for the Issue Certificates database are `cert.db`, `cert.lck`, `cert.01`, and the `cert.rfl` directory.

1. Make sure the Organizational CA is functional.
2. Back up the Organizational CA.
3. Delete the Organizational CA object.
4. Restore the Organizational CA to the desired server.

IMPORTANT: Be sure to protect your backup media.

3.1.10 Validating the Organizational CA's Certificates

If you suspect a problem with a certificate or think that it might no longer be valid, you can easily validate the certificate by using iManager. Any certificate in the eDirectory tree can be validated, including certificates issued by external CAs.

The certificate validation process includes several checks of the data in the certificate as well as the data in the certificate chain. A certificate chain is composed of a root CA certificate and, optionally, the certificates of one or more intermediate CAs.

A result of Valid means that all certificates in the certificate chain were found to be valid. Certificates are considered valid if they pass a predefined set of criteria including whether the current time is within the validity period of the certificate, whether it has not been revoked, and whether it has been signed by a CA that is trusted. Only those certificates with a CRL distribution point extension or an OCSP AIA extension are checked for revocation.

A result of Invalid means that one or more certificates in the certificate chain were found to be invalid or their validity could not be determined. Additional information is provided for these certificates, indicating which certificate is considered invalid and why. Click *Help* for more information about the reason.

To validate a certificate:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Configure Certificate Authority*.
4. Click *Certificates*, then select the public key certificate or the self signed certificate.
5. Click *Validate*.

   The status of the certificate is reported in the *Certificate Status* field. If the certificate is not valid, the reason is given.
6. Click *OK*.

3.1.11 Deleting the Organizational CA

Deleting the Organizational CA object should be done only if absolutely necessary or if you are restoring the Organizational CA from a backup (see "Restoring an Organizational CA" on page 33). The only safe way to delete the object is to do a backup first so that it can be restored later.

However, there are times when the Organizational CA must be deleted and not restored. For example, when merging trees, only one Organizational CA can be in the resulting tree; the other CA must be deleted. Or, when the Organizational CA's host server is irreparably damaged and no backup of the CA or the NICI configuration was made, the only option remaining is to delete the CA and to begin again.

To delete the Organizational CA object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. Back up the self-signed certificate without the private key.

4 Create a Trusted Root certificate using the self-signed certificate in the CN=trusted roots.CN=security container. For more information, see

5 On the *Roles and Tasks* menu, click *Directory Administration* > *Delete Object*.

6 Browse to and click the Organizational CA object.

7 Click *OK*.

3.1.12 Rolling Over an Organizational CA

Two important issues that must be considered when replacing the Organizational Certificate Authority (CA) certificates are:

- The types of certificates that are being managed
- The reason that the CA is being replaced

Server Certificate objects (KMOs) contain both the public key certificate for the server and the Trusted Root certificate with which the public key certificate was signed.

User certificates are stored as attributes on the user object and are not paired with the Trusted Root that signed them. Therefore, when the trusted root certificate is replaced, server certificates are still valid because the trusted root is still accessible. However, user certificates are immediately invalid unless the Trusted Root certificate is placed in the Trusted Roots container where certificate validation can find it.

There are three reasons to replace the CA:

- The CA has reached the end of its validity (the CA is expiring).
- The CA has been compromised.
- You want to replace the CA certificate for some other reason (a stronger key is desired, a new security policy has made it necessary, you want to have an externally signed CA, etc.).

If the CA is expiring, the certificates that the CA signed are also going to expire. After replacing the CA, each of the signed certificates should be re-created with the new CA.

If the CA has been compromised, then replacing the CA invalidates the user certificates that were signed by the old CA. You can easily replace them by running the Create Default Certificates task in iManager. All certificates that are created by default by Certificate Server are re-created with the new CA. Any certificates that were created in a custom manner need to be manually re-created with the new CA. For more information on creating default certificates through, see Section 3.2.2, “Creating Default Server Certificate Objects,” on page 37.

If you want to re-create the CA for some other reason, then storing the trusted root certificate in the Trusted Roots container keeps user certificates valid until you have a chance to re-create them at your convenience.

To replace the trusted root certificate:

1 Back up the current CA in case you want to recover it later.

2 Export the Trusted Root certificate that has been used to create the certificates. In older systems, this is most likely the self-signed certificate.

   Recently, the ability has been added to externally sign the CA certificate. If the CA is externally signed, export the public key certificate. All certificates in the chain must have their own object in the Trusted Roots container.

   If the CA has not been compromised, create a Trusted Root certificate in the Trusted Roots container. This ensures that user certificates are still valid until they can be replaced.

3 Delete the old CA. For information on deleting the Organizational CA, see Section 3.1.11, “Deleting the Organizational CA,” on page 35.

4 Create a new CA. For information on creating a new Organizational CA, see Section 2.2, “Creating an Organizational Certificate Authority Object,” on page 20.

5 If necessary, re-create server certificates by using the Create Default Certificate task in iManager. For information on creating default certificates through iManager, see Section 3.2.2, “Creating Default Server Certificate Objects,” on page 37.

Re-create other server certificates that are not generated by default.

6 If necessary, re-create user certificates by using the Create User Certificate task in iManager or by viewing the user properties, viewing certificates, and clicking *New*.

3.2 Server Certificate Object Tasks

3.2.1 Creating Server Certificate Objects

This task is described in Section 2.4, “Creating a Server Certificate Object,” on page 23.

3.2.2 Creating Default Server Certificate Objects

The Certificate Server installation creates default Server Certificate objects.

- SSL CertificateDNS - *server_name*
- A certificate for each IP address configured on the server (IPAG*xxx.xxx.xxx.xxx - server_name*)
- A certificate for each DNS name configured on the server (DNSAG*www.example.com - server_name*)

NOTE: eDirectory 8.8 SP8 does not automatically create SSL CertificateIP. SSL Certificate DNS contains all the IPs listed in the Subject Alternative Name.

When you attempt to create or repair the default certificates using the PKI iManager plug-in, the SSL CertificateIP certificate will not be created or repaired by default. However, a check box has been provided in the plug-in interface which you can select to override the default behavior and force the creation/repair of the SSL CertificateIP certificate.

If these certificates become corrupt or invalid for some reason, or if you just want to replace the existing default certificates, you can use the Create Default Server Certificates Wizard, as described in the following procedure:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Create Default Certificates*.
4. Browse for and select the server or servers that you want to create default certificates for, then click *Next*.
5. Select *Yes* if you want to overwrite the existing default server certificates or select *No* if you want to overwrite the existing default server certificates only if they are invalid.
6. (Single Server only) If you want to use the existing default IP address, select that option. If you want to use a different IP address, select that option and specify the new IP address.
7. (Single Server only) If you want to use the existing DNS address, select that option. If you want to use a different DNS address, select that option and specify the new DNS address.
8. Click *Next*.
9. Review the summary page, then click *Finish*.

If you want more control over the creation of the Server Certificate object, you can create the Server Certificate object manually. For more information, see Section 2.4.1, "Manually Creating a Server Certificate Object," on page 23.

3.2.3 Importing a Public Key Certificate into a Server Certificate Object

You import a public key certificate after you have created a certificate signing request (CSR) and the Certificate Authority (CA) has returned the signed public key certificate to you. This task applies when you have created a Server Certificate object by using the Custom option with the External CA signing option.

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. These files can be binary, DER-encoded files (`.der`, `.cer`, `.crt`., `.p7b`) or they can be textual, Base64-encoded files (`.cer`, `.b64`).

If the file has multiple certificates in it, it must be in PKCS #7 format in order to be imported into a Server Certificate object. Additionally, the file must contain all of the certificates to be imported into the object (the root-level CA certificate, any intermediate CA certificates, and the server certificate).

If the CA returns multiple files to you as a result of signing the certificate, each file contains a different certificate that must be imported into the Server Certificate object. If there are more than two files (one for the root-level CA, one or more for the intermediate CAs, and one for the server certificate), these files must be combined into a PKCS #7 file in order to be imported into a Server Certificate object.

There are several ways to create a PKCS #7 file. One way is to import all of the certificates into Internet Explorer. After they have been imported, the server certificate and all of the certificates in the certificate chain can be exported in PKCS #7 format by using Internet Explorer. For more information on how to do this, see Section 4.4.2, "External CAs," on page 81.

Some CAs do not return a root-level CA certificate along with the server certificate. In order to obtain the root-level CA certificate, contact the CA provider directly or call Technical Support.

To import the certificates into a Server Certificate object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *Server Certificates*.
4. Click *Import* next to the Server Certificate object you want to modify.
5. Browse for and select the certificate data file.
6. Browse for and select the trusted root data file.

   If all certificates are contained in a single file, leave this field blank.
7. Click *OK*.

3.2.4 Exporting a Trusted Root or Public Key Certificate

You export a certificate to a file for the following reasons:

- A client (such as an Internet browser) can use it to verify the certificate chain sent by a cryptography-enabled application.
- To provide a backup copy of the file.

You can export the certificate in two file formats: DER-encoded (`.der`) and Base64-encoded (`.b64`). The `.crt` extension can also be used for DER-encoded certificates. You can also export to the system clipboard in Base64 format so that the certificate can be pasted directly into a cryptography-enabled application.

To export a trusted root or public key certificate:

1. Launch iManager.
2. Log in to the eDirectory tree as a user with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *Server Certificates*.
4. Select the Server Certificate object the particular application is configured to use.
5. Click *Export*.

   This opens a wizard that helps you export the certificate to a file.
6. Use the drop-down list to specify which certificate to export.

7 Choose not to export the private key.

8 Select an export format (binary DER or text encoded base64), then click *Next*.

9 Click *Save the exported certificate to a file* and save the file to a location of your choice.

10 Click *Close* > *Close* > *OK*.

11 Use the file as needed.

For example, if you want to install a trusted root certificate in an Internet Explorer browser, double-click the file. This initiates a wizard that will accept the CA as a trusted root. Accepting the CA as a trusted root means that the browser automatically accepts SSL connections with services that use certificates issued by this CA.

3.2.5 Deleting a Server Certificate Object

You should delete a Server Certificate object if you suspect that the private key has been compromised, if you no longer want to use the key pair, or if the trusted root in the Server Certificate object is no longer trusted.

IMPORTANT: After the Server Certificate object is deleted, you cannot recover it unless you have previously made a backup. Before you delete this object, make sure that no cryptography-enabled applications still need to use it.You can re-create a Server Certificate object, but you will need to reconfigure any applications that referenced the old object.

To delete a Server Certificate object:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *Server Certificates*.

4 Select the Server Certificate object you want to delete.

5 Click *OK* to delete the object.

3.2.6 Viewing a Server Certificate Object's Properties

In addition to the eDirectory rights and properties that are viewable with any eDirectory object, you can also view properties specific to the Server Certificate object, including the properties of the public key certificate and the Trusted Root certificate associated with it, if they exist.

To view a Server Certificate object's properties:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *Server Certificates*.

4 Click the nickname of the Server Certificate object you want to view.

5 To view the certificate chain, click the plus sign (+) in front of the certificate’s nickname to expand the view.

6 Click *Cancel*.

3.2.7 Viewing a Server Certificate Object's Public Key Certificate Properties

To view a Server Certificate object's public key certificate properties:

1. Launch iManager.
2. Log in to the eDirectory tree as a user with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.
4. Browse to and click the Server Certificate object you want to view.
5. Click *OK*.
6. Click Public Key Certificate.
   - If a public key certificate is installed, the property page displays the subject's fully typed name, the issuer's fully typed name, and the validity dates of the public key certificate.
   - If the public key certificate has not yet been installed, the property page indicates this.
7. To view the certificate chain, click the plus sign (+) in front of the certificate's nickname to expand the view.
8. To view additional information about a public key certificate, click the certificate's nickname to view the Details page.

   The Details page has information contained in the public key certificate.
9. Click *Close* > *Cancel*.

3.2.8 Viewing a Server Certificate Object's Trusted Root Certificate Properties

To view a Server Certificate object's Trusted Root certificate properties:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.
4. Browse to and select the Server Certificate object you want to view.
5. Click *OK*.
6. Click *Trusted Root Certificate*.
   - If a Trusted Root certificate is installed, the property page displays the subject's fully typed name, the issuer's fully typed name, and the validity dates of the trusted root certificate.
   - If the Trusted Root certificate has not yet been installed, the property page indicates this.
7. To view the certificate chain, click the plus sign (+) in front of the certificate's nickname to expand the view.
8. To view additional information about a Trusted Root certificate, click the certificate's nickname to view the Details page.

   The Details page has information contained in the trusted root certificate.
9. Click *Close* > *Cancel*.

3.2.9 Backing Up a Server Certificate Object

NetIQ Certificate Server allows you to store certificates signed by third-party certificate authorities in server certificate objects. Often these certificates cost a significant amount of money. Unfortunately, if an unrecoverable failure happens on the server that owns the certificates, the server certificate object can no longer be used. In order to protect against such failures, you might want to back up server certificates signed by external CAs and their associated private keys. Then, if a failure should occur, you can use the backup file to restore your server certificate object to any server in the tree that has Certificate Server version 2.21 or higher installed.

NOTE: The ability to back up a Server Certificate object is only available for objects created with Certificate server version 2.21 or later. In previous versions of Certificate Server, the server's private key was created in a way that made exporting it impossible.

The back up file contains the server's private key, public key certificate, trusted root certificate, and any intermediate CA certificates stored. This information is stored in PKCS #12 format (also known as PFX).

A server certificate object should be backed up when it is working properly.

To backup a Server Certificate object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.
4. Browse to and click the Server Certificate object you want to back up.
5. Click *OK*.
6. Click the *Certificates* tab.
7. Click either the Trusted Root certificate or the public key certificate. Both certificates are written to the file during the backup operation.
8. Click *Export*.

   This opens a wizard that helps you export the certificates to a file.
9. When asked whether to export the private key, select *Yes*, then click *Next*.
10. Specify a password with 6 or more alphanumeric characters to use in encrypting the PFX file.
11. Click *Next*.
12. Click *Save the exported certificate to a file*. Select the filename and the location for the backup file.
13. Click *Close*.

    The encrypted backup file is written to the location specified. It is now ready to be stored in a secure location for emergency use.

IMPORTANT: The exported file should be put on a diskette or some other form of backup media and stored in a secure place. The password used to encrypt the file should be committed to memory or stored in a vault to ensure that it is available when needed, but inaccessible to others.

3.2.10 Restoring a Server Certificate Object

If the Server Certificate object has been deleted or corrupted, or if the server that owned the Server Certificate object has suffered an unrecoverable failure, the object can be restored to full operation using a backup file created as described in "Backing Up a Server Certificate Object" on page 42.

The ability to restore a Server Certificate object is only available in Certificate Server version 2.21 or later.

If you were unable to make a backup of the server certificate object, the server certificate object might still be usable if NICI 2.x is installed on the server and a backup was made of the NICI configuration information. For information on how to back up and restore the NICI configuration files, see the "Backing Up and Restoring NICI" (https://www.netiq.com/documentation/nici27x/nici_admin_guide/data/bwf6d4c.html) section in the *Novell International Cryptographic Infrastructure 2.7 Administration Guide*.

To restore the Server Certificate object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. Delete the old server certificate object.
4. On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create Server Certificate*.

   This opens the Create a Server Certificate Wizard that creates the object.
5. In the wizard, specify the server that should own the server certificate object, and specify the certificate nickname of the server certificate. The server must have Certificate Server version 2.21 or higher installed and be up and running.
6. Select the *Import* option, then click *Next*.
7. Browse for and select the backup file, enter the backup file password, then click *Finish*.

The server's private key and certificates have now been restored and the Server Certificate object is fully functional. The backup file can be stored again for future use if desired.

IMPORTANT: Be sure to protect your backup media.

3.2.11 Server Certificate Objects and Clustering

You can set up Server Certificate objects in a clustered environment to ensure that your cryptography-enabled applications that use Server Certificate objects always have access to them. Using the backup and restore feature for Server Certificate objects, you can duplicate the object's keying material from one node in the cluster to all nodes. Using this process for keying material signed by an external CA saves you money by allowing you to duplicate the keying material for one server certificate rather than requiring new keying material for every node in the cluster.

To set up server certificates to work in a clustered environment:

1. Create a server certificate on a server in the cluster, using either the Organizational CA or an external CA of your choice. See Section 2.4, "Creating a Server Certificate Object," on page 23.

   When you create the server certificate objects, the Common Name (CN) portion of the certificate's subject name should be an IP or DNS name that is specific to the service. Otherwise, you receive a browser warning message indicating that the IP or DNS name on the URL does not match that in the certificate.

If different services have different IP or DNS addresses, you need to create a server certificate for each service.

2 Back up the keying material for this server certificate object and restore it by creating a Server Certificate object with the same key pair name as the one you created in Step 1 on all remaining servers in the cluster.

See "Backing Up a Server Certificate Object" on page 42.

3.2.12 Validating a Server Certificate

If you suspect a problem with a certificate or think that it might no longer be valid, you can easily validate the certificate by using iManager. Any certificate in the eDirectory tree can be validated, including certificates issued by external CAs.

The certificate validation process includes several checks of the data in the certificate as well as the data in the certificate chain. A certificate chain is composed of a root CA certificate and, optionally, the certificates of one or more intermediate CAs.

A result of Valid means that all certificates in the certificate chain were found to be valid. Certificates are considered valid if they pass a predefined set of criteria including whether the current time is within the validity period of the certificate, whether it has not been revoked, and whether it has been signed by a CA that is trusted. Only those certificates with a CRL distribution point extension or an OCSP AIA extension are checked for revocation.

A result of Invalid means that one or more certificates in the certificate chain were found to be invalid or their validity could not be determined. Additional information is provided for these certificates, indicating which certificate is considered invalid and why. Click Help for more information about the reason.

To validate a certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *Server Certificates*.

4 Select the Server Certificate object you want to validate.

5 Click *Validate*.

The status of the certificate is provided in the *Certificate Status* field. If the certificate is not valid, the reason is given.

3.2.13 Revoking a Trusted Root or Self Signed Certificate

You might find it necessary to revoke a certificate if the key or the CA becomes compromised, if the certificate has been superseded by another certificate, if the certificate is removed from the CRL, cessation of operation, etc.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.

4 Browse to and click the Server Certificate object you want to modify.

5 Click *OK*.

6 Click the *Certificates* tab.

7 Click *Trusted Root Certificate* or *Self Signed Certificate*.

8 Select the certificate, then click *Revoke*.

This starts the Revoke Certificate Wizard. Follow the prompts to revoke the certificate.

9 Click *Finish*.

3.2.14 Moving a Server Certificate Object to a Different Server

You can move a Server Certificate object from one server to another by using the backup and restore procedures outlined in "Backing Up a Server Certificate Object" on page 42 and "Restoring a Server Certificate Object" on page 43.

1 Make sure the Server Certificate object is functional.

2 Back up the Server Certificate object.

3 Restore the Server Certificate object to the desired server.

IMPORTANT: Be sure to protect your backup media.

3.2.15 Replacing a Server Certificate Object's Keying Material

The private key and certificates in the server certificate object can be replaced. They should only be replaced using an internally generated PFX file created during a backup of a server certificate object. Externally generated PFX files can also be used if they contain the private key, the server certificate, and the entire certificate chain. The key and certificates in the file need not match the ones in the object; the data in the file overwrites the key and certificates in the object.

Replacing the private key and certificates in the server certificate object is a serious matter. If the key and certificates do not exactly match the ones in the object, it is the same as deleting the current server certificate object and creating a new one. See the section "Deleting a Server Certificate Object" on page 40 for more information on the consequences of deleting the object.

If the key and certificates do match the ones in the object, replacing the keying material has no effect except to regenerate a few attributes used by the Secure Authentication Services (SAS) and NILE services.

To replace the keying material on the Server Certificate object:

1 As a precaution, back up the server certificate object with the private key. See "Backing Up a Server Certificate Object" on page 42.

2 Launch iManager.

3 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

4 On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.

5 Browse to and select the Server Certificate object you want to modify.

6 Click *OK*.

7 Click the *Certificates* tab.

8 Click *Trusted Root Certificate* or *Self Signed Certificate*.

   The operation can be started from either page. It replaces both certificates as well as the private key and any other certificates in the certificate chain.

9 Select the certificate, then click *Replace*.

   This opens a wizard that helps you specify the PFX (backup) file.

10 Browse for and select the backup file, enter the backup file password, then click *OK*.

The server's private key and certificates have now been replaced and the server certificate is fully functional. The backup file should be stored again for future use if desired.

IMPORTANT: Be sure to protect your backup media.

3.3 User Certificate Tasks

3.3.1 Creating User Certificates

This task is described in "Creating a User Certificate" on page 24.

3.3.2 Creating User Certificates in Bulk

This feature allows you to create user certificates for multiple users at the same time, using one sequence of operations.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Server* > *Create User Certificate*.

   This opens a wizard that helps you create the user certificate.

4 Browse for and select all users you want to create a user certificate for.

5 Follow the wizard prompts to create the certificate for each user. For specific information on the wizard pages, click *Help*.

3.3.3 Importing a Public Key Certificate into a User Object (with or without the Private Key)

You can import any public key certificate into a user object (for example, a certificate signed by a third-party certificate authority). This certificate can appear as one of two types of files:

- **DER:** Contains a public key certificate only.
- **PFX or PKCS#12:** Contains a public key certificate as well as a private key.

After it is imported, the certificate is stored in the User object and appears on the list of certificates available.

NOTE: When importing a PKCS#12 certificate, only the public key certificate and private key are stored on the User object. No other certificates are stored. Other certificates in the user's certificate chain should probably be stored in the CN=Trusted Roots.CN=Security container (create a new Trusted Root object for each certificate in the chain).

To import a Public Key Certificate into a User object:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.
4. Browse for and select a User object to import the public key certificate into.
5. Click *New*.
6. Specify a nickname for the user certificate.

   The nickname should be unique and should help you identify the certificate. You can enter up to 64 characters in the *Certificate Nickname* field.
7. Select the import creation method, then click *Next*.
8. Browse for and select the certificate to import, then click OK.
9. (Conditional) If you are importing a certificate with a private key, enter the password for the private key, then click Next.
10. Click *Finish*.

    This stores the certificate in the User object, and the certificate appears on the list of certificates available to this user.

3.3.4 Viewing a User Certificate's Properties

In addition to the eDirectory rights and properties that are viewable with any eDirectory object, you can also view properties specific to the user certificate, including the issuer, the certificate status, the private key status, and the validation period.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.

4 Browse for and select a User object whose certificate properties you want to view.

5 To view the certificate chain, click the plus sign (+) in front of the certificate's nickname to expand the view.

6 Click the nickname of the certificate to view its details.

7 Click *Close* when you are done viewing.

3.3.5 Exporting a User Certificate

In order to exchange secure e-mail with another person, you must first have the other person's public key certificate. One way of obtaining that certificate is to export it using iManager. The other person's certificate can also be obtained by using LDAP or e-mail.

To export your own or any other user's public key certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.

4 Browse for and select a User object whose certificate you want to export.

5 Select the certificate, then click *Export*.

This opens a wizard that helps you export the user certificate to a file. If you are logged in as the user that owns the certificate, select *No* when asked if you want to export the private key. See "Exporting a User Certificate and Private Key" on page 48.

6 If you want to export the private key, select *Export private key* and provide a password to protect the private key.

7 Select an export format if you are not exporting the private key, then click *Next*.

8 Click *Save the exported certificate to a file* and save the file to a location of your choice.

9 Click *Close* > *Close*.

3.3.6 Exporting a User Certificate and Private Key

In order to use a certificate for secure e-mail, authentication, or encryption, both the private key and the certificate must be available to the cryptography-enabled application. You must export the user certificate and private key and place it in a location that the application has access to in order for the application to use them.

The private keys in a user's object belong to that user. Only someone logged in as that user can export the private key. No other user, not even the network administrator, has rights to export another user's private key.

To export your own private key and certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as the user who owns the certificate.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.

4 Browse for and select a User object whose certificate you want to export.

5 Select the certificate, then click *Export*.

   This opens a wizard that helps you export the user certificate to a file.

6 Select *Export private key*, provide a password to protect the private key, then click *Next*.

7 (Optional) Click *Export the Certificate into the Browser*.

8 Click *Close* > *Close*.

   The encrypted file is written to the location specified. It is now ready to be imported into a cryptography-enabled application.

IMPORTANT: The exported file can be kept to provide a backup. If so, it should be stored in a secure place. The password used to encrypt the file should be committed to memory or stored in a safe place to ensure that it is available when needed, but inaccessible to others.

3.3.7 Validating a User Certificate

If you suspect a problem with a certificate or think that it might no longer be valid, you can easily validate the certificate by using iManager. Any certificate in the eDirectory tree can be validated, including certificates issued by external CAs.

The certificate validation process includes several checks of the data in the certificate as well as the data in the certificate chain. A certificate chain is composed of a root CA certificate and, optionally, the certificates of one or more intermediate CAs.

A result of Valid means that all certificates in the certificate chain were found to be valid. Certificates are considered valid if they pass a predefined set of criteria including whether the current time is within the validity period of the certificate, whether it has not been revoked, and whether it has been signed by a CA that is trusted. Only those certificates with a CRL distribution point extension or an OCSP AIA extension are checked for revocation.

A result of Invalid means that one or more certificates in the certificate chain were found to be invalid or their validity could not be determined. Additional information is provided in these cases about which certificate is considered invalid and why. Click Help for more information about the reason.

To validate a certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.

3 On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.

4 Browse for and select a User object whose certificate you want to validate.

5 Select the user certificate you want to validate.

6 Click *Validate*.

   The status of the certificate is provided in the *Certificate Status* field. If the certificate is not valid, the reason is given.

NOTE: If the user certificate was signed by a third-party CA, the certificate chain must be in the Trusted Roots container in the Security container (CN=Trusted Roots.CN=Security) for the validation to succeed. Typically, the certificate chain consists of a single, root-level CA or it consists of an Intermediate CA and a root-level CA. The name of the Trusted Roots container must be Trusted Roots

and each certificate in the chain must be stored in its own Trusted Root object. For instructions on how to create a Trusted Roots container and Trusted Root objects, see "Creating a Trusted Root Container" on page 57 and "Creating a Trusted Root Object" on page 57.

When validating user certificates or intermediate CA certificates signed by external CAs, the external CA's certificate must be stored in a Trusted Root object in order for the certificate validation to be successful. The Trusted Root object must be in a Trusted Root Container named Trusted Roots and it must be located in the Security container.

3.3.8 Revoking a User Certificate

You might find it necessary to revoke a certificate if the key or the CA becomes compromised, if the certificate has been superseded by another certificate, if the certificate is removed from the CRL, cessation of operation, etc.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.
4. Browse for and select a User object whose certificate you want to validate.
5. Select the user certificate you want to revoke.
6. Click *Revoke*.

   This starts the Revoke Certificate Wizard. Follow the prompts to revoke the certificate.
7. Click *Finish*.

3.3.9 Deleting a User Certificate and Private Key

If a user certificate has become invalid or you suspect the private key has been compromised in some way, you might need to delete the user certificate and private key.

Before you delete a user certificate and private key, you should revoke the user certificate. See Section 3.3.8, "Revoking a User Certificate," on page 50.

To delete a user certificate and private key:

1. Launch iManager.
2. Log in to the eDirectory tree as the user who owns the certificate or as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *NetIQ Certificate Access* > *User Certificates*.
4. Browse for and select a User object whose certificate you want to delete.
5. Select the user certificate you want to delete.
6. Click *Delete*.

3.4 X.509 Certificate Self-Provisioning

This section describes the X.509 self-provisioning feature.

3.4.1 Overview

When you create an X.509 certificate, there are many important pieces of information that must be identified and substantiated before the certificate authority (CA) issues the certificate. Two of the most important tasks are:

- Verifying the identity of the certificate's subject (verifying the identity of the person or object the certificate is for).
- Verifying the appropriateness of the subject name in the certificate (verifying that the subject name correctly represents the identity of the person or object the certificate is for).

These two tasks can be very time-consuming and are often performed by a separate administrative person or group.

NetIQ Certificate Server has always leveraged the secure identity management capabilities of eDirectory to reduce the time and effort needed to perform these verifications. iManager allows an administrator to create user certificates in bulk; that is, to create a certificate for a large number of users at one time. The CA checks that the identity of the certificate is tied to the eDirectory account, which verifies the identity of the certificate's subject; however, the CA has not verified the appropriateness of the subject name in the certificate. Because of this, creating certificates with NetIQ Certificate Server has always required that the person or software have administrative rights to the Organizational CA.

Self-provisioning allows a user or server to generate certificates without having administrative rights to the Organizational CA and without intervention of a separate administrative person or group, and still maintain the security of the CA.

NetIQ Certificate Server verifies the identity of the certificate's subject by checking that the identity of the certificate is tied to the eDirectory account. The CA also verifies the appropriateness of the subject name in the certificate by checking against information in eDirectory. This allows the Organizational CA to leverage the security identity management capabilities of eDirectory to reduce administrative tasks while maintaining the security of the CA.

3.4.2 User Self-Provisioning

In the past, creating a user certificate required administrative rights to the CA as well as rights to attributes on the User object. With user self-provisioning, administrative rights to the CA are not necessary; however, Read (R) and Write (W) rights to the userCertificate, NDSPKI:UserCertificateInfo, and SAS:SecretStore attributes are still necessary.

If the person requesting the creation of the certificate has administrative rights to the CA, the certificate creation is not affected by whether or not user self-provisioning is enabled. If the person requesting the creation of the certificate does not have administrative rights to the CA, the subject name in the request is compared to the user's eDirectory DN and any values in the sasAllowableSubjectNames attribute.

If the subject name matches, the CA checks to ensure that any Subject Alternative Names are appropriate. The CA does this by checking that there is not more than one Subject Alternative Name. If the name exists, it must be of type email name and it must match a configured email name on the User object. If all these checks succeed, the CA does not require administrative rights to the CA in order to create the certificate.

To use user self-provisioning:

1. Ensure that you have eDirectory 8.8 and the NetIQ Certificate Server 3.2.2 or later plug-in for iManager installed.

   Both eDirectory 8.8 and the NetIQ Certificate Server 3.2.2 plug-in for iManager are included with Open Enterprise Server (OES) 2 and are installed automatically when you select any of the eDirectory-required components during the OES 2 installation.

2. Enable user self-provisioning
   - **2a** Launch iManager.
   - **2b** Log in to the eDirectory tree as an administrator with administrative rights to the Organizational CA.
   - **2c** On the *Roles and Tasks* menu, click *NetIQ Certificate Server > Configure Certificate Authority*.
   - **2d** Select *Enable user self-provisioning*.
   - **2e** Click *OK*.
3. Set up inherited rights for users by enabling the iManager "[this]" object:
   - **3a** Log in to iManager as an iManager administrator.
   - **3b** Click the *Configure* icon.
   - **3c** Click *iManager Server > Configure iManager*.
   - **3d** Click the *Misc* tab.
   - **3e** Select *Enable "[this]"*.
   - **3f** Click *Save*.

     Next, you need to add inherited rights.
4. Log in to iManager as a Certificate Authority administrator.
5. On the *Roles and Tasks* menu, click *Rights > Modify Trustees*.
6. Browse for and select the object you want the rights to be inherited from (for example, the root of the tree or a container), then click *OK*.
7. Click *Add Trustee*, select the *"[this]"* object, then click *OK*.
8. Click *Assigned Rights*.
9. Click *Add Property*.
10. Select *Show all properties in schema*.
11. Select the `userCertificate` attribute, then click *OK*.
12. Select *Read* and *Write* rights.
13. Select *Inherit*.
14. Repeat Step 6 through Step 10 for the other attributes (`NDSPKI:UserCertificateInfo` and `SAS:SecretStore`).
15. Click *Done > OK*.

3.4.3 Server Self-Provisioning

In past, creating a server certificate required administrative rights to the CA as well as administrative rights to the context the server certificate was to be created in. With server self-provisioning, administrative rights to the CA are not necessary; however, administrative rights to the context the server certificate was created in are still necessary.

To create the server certificate, you must have the administrative rights to the CA. The certificate creation is not affected by whether or not server self-provisioning is enabled. In case you do not have the required administrative rights to the CA, enable the *Require read rights to operate the CA* option to operate the CA from the *Configure Certificate Authority* task in iManager. The administrative rights to the CA are not required if any one of the following are true:

- The subject name in the request is compared to the server's eDirectory DN and any IP or DNS addresses as determined by a DNS or eDirectory SLP lookup. If the subject name matches either, the CA does not require administrative rights to the CA in order to create the certificate.
- Non CN components of the subject name matches non CN components of CA certificate's subject name.
- Subject alternative name only has IP-address/DNS name that is verified again by CA through reverse DNS lookup.

The servers are not granted write rights to the CA's *NDSPKI:Private Key* attribute by default. If *Require write rights to operate the CA* is enabled from the *Configure Certificate Authority* task in iManager, you should grant servers the write rights to the CA's *NDSPKI:Private Key* attribute.

NOTE: Be aware that when PKI Health Check runs on a server with server self-provisioning enabled, your server's server certificates might be automatically created (if none exist) or replaced (if they are expired). For more information, see Section 3.10, "PKI Health Check," on page 73

To use server self-provisioning:

1. Ensure you have eDirectory 8.8 and the NetIQ Certificate Server 3.2.2 or later plug-in for iManager installed.

   Both eDirectory 8.8 and the NetIQ Certificate Server 3.2.2 plug-in for iManager are included with OES 2 and are installed automatically when you select any of the eDirectory-required components during the OES 2 installation.

2. Enable server self-provisioning:

   2a. Launch iManager.

   2b. Log in to the eDirectory tree as an administrator with administrative rights to the Organizational CA.

   2c. On the *Roles and Tasks* menu, click *NetIQ Certificate Server > Configure Certificate Authority*.

   2d. Select *Enable server self-provisioning*.

   2e. Click *OK*.

3.4.4 Certificate Self-Provisioning and the Issue Certificate Task

The Issue Certificate task allows the creation of a certificate by using a PKCS#10 certificate signing request (CSR). This task allows the user to create a certificate that is not tied to any eDirectory object. If the person requesting the creation of the certificate has administrative rights to the CA, the certificate creation is not affected. If the person requesting the creation of the certificate does not have administrative rights to the CA, the certificate request is treated as a user self-provisioning request,

but the person does not need to have rights to the attributes userCertificate, NDSPKI:UserCertificateInfo, and SAS:SecretStore attributes on the object. This is because the certificate is not stored in eDirectory, so rights to the object are not needed.

User self-provisioning must be enabled for a user to issue certificates without having administrative rights to the CA. Complete Steps 1 through 3 of Section 3.4.2, "User Self-Provisioning," on page 51.

For information on the Issue Certificate task, see Section 3.1.2, "Issuing a Public Key Certificate," on page 30.

3.5 Using eDirectory Certificates with External Applications

This option is not supported on OES 1.

Some customers use non-eDirectory applications that require X.509 certificates and keys (for example, Apache or OpenSSL). Most of these applications are configured out of the box with self-signed (no value) certificates, which are meant only to provide a temporary solution until the application can be configured with real X.509 certificates and keys.

Unfortunately, many administrators do not replace these self-signed certificates, often because it is too time-consuming or too difficult. In addition, X.509 certificates are designed to expire regularly, so replacing them on a regular basis is an ongoing administrative task.

The following sections describe the solution to this problem:

- Section 3.5.1, "PKI Health Check Functionality," on page 54
- Section 3.5.2, "Configuring the SAS:Service Object to Export eDirectory Certificates," on page 55

3.5.1 PKI Health Check Functionality

In response to customer requests to provide non-eDirectory applications with X.509 certificates, the PKI Health Check code within NetIQ Certificate Server now provides the capability to automatically export X.509 certificates and keys to the file system, enabling non-eDirectory applications to take advantage of eDirectory-minted certificates and eDirectory-managed certificates.

When the PKI Health Check runs, it automatically overwrites any existing certificates, including the certificates' private keys. However, to ensure that no valid certificates and private keys are deleted, the PKI Health Check determines whether the existing certificates and keys are the same as those configured in eDirectory. If they are different than those configured in eDirectory, the PKI Health Check creates a backup of these files before overwriting them. This ensures that certificates that have been acquired from an external source (for example, VeriSign*) are not deleted.

After a configuration has been created for the server on the SAS:Service Object, keys and certificates associated with the specified server are automatically exported to the file system. If the keys and certificates are replaced or updated in eDirectory (for example, if the Server Certificate object is deleted and a new one is created with the same name), the new keys and certificates are automatically exported to the file system the next time PKI Health Check runs.

NOTE: The PKI Health code within NetIQ Certificate Server runs once every time NetIQ Certificate Server loads/reloads. You can use any of the following methods to reload the NetIQ Certificate Server:

- Restart the server
- Restart eDirectory
- Unload and load PKI Server manually

- Run an eDirectory repair (NDSRepair)

  NetIQ Certificate Server shuts down during the repair and reloads after the eDirectory repair is finished.

For more information on the PKI Health Check, see Section 3.10, "PKI Health Check," on page 73.

Before the PKI Health Check can automatically export X.509 certificates and keys to the file system, the SAS:Service Object must be configured. This is because the PKI Health Check reads the configuration on the SAS:Service Object. For information on how to configure the SAS:Service Object, see Section 3.5.2, "Configuring the SAS:Service Object to Export eDirectory Certificates," on page 55.

3.5.2 Configuring the SAS:Service Object to Export eDirectory Certificates

Before an eDirectory Server Certificate can be exported to the file system, a configuration must first be created for the server on the SAS:Service Object. This can be done either automatically or manually, depending on what eDirectory server you are using. Only OES 2 Linux servers can be automatically configured during installation to create this configuration; on all other eDirectory servers, you must manually create this configuration. The following sections further explain these options:

Manually Configuring the SAS:Service Object to Enable Use of eDirectory Certificates

If you are not using OES 2 as your eDirectory server, you must manually configure the SAS:Service Object in order to export eDirectory certificates. This configuration must specify the Server Certificate name. If multiple server certificates need to be exported, you can simply create multiple configurations. You can export the same certificate to a different file path, or you can export a different certificate to a different file path.

NOTE: Each configuration must use unique file paths in order to avoid file collisions. The Public key path and the Private key path must be unique and different from each other and from any other configuration.

To create a configuration on the SAS:Service object:

1. In iManager, in the *Roles and Tasks* view, click *NetIQ Certificate Access*.
2. Click *SAS Service Object*.
3. On the SAS Service Object page, click the *Browse* icon.
4. Browse to and select the SAS:Service object where you want to create the configuration.
5. Click the SAS:Service object.
6. Click *New*.

   The Server Certificate Synchronization window is displayed.
7. In the *Certificate* field, browse for and select the certificate you want to export.
8. In the *Public key path* field, specify the path where the application will find and use the certificate. For example: `C:/novell/nds/servercert.pem`.

9 In the *Private key path* field, specify the path where the application will find and use the certificate's private key. For example: `C:/novell/nds/serverkey.pem`.

10 Select the key type that you are going to use. If you are running OpenSSL, select PKCS#8. If you are running Apache, select PKCS#1.

11 Click *OK*.

The configuration is created. The name, path, key path and key type are displayed.

To create another configuration, repeat Step 6 through Step 11.

If you are using a Linux server that is running OES 2 or later as your eDirectory server, then you can automatically configure the server to create a configuration on the SAS:Service Object. For more information on how to do this, see "Automatically Configuring the SAS:Service Object to Enable Use of eDirectory Certificates (OES 2 Only)" on page 56.

Automatically Configuring the SAS:Service Object to Enable Use of eDirectory Certificates (OES 2 Only)

When installing OES 2 on Linux, the YaST installer provides a configuration screen that allows you to specify whether you want to automatically configure the server to export eDirectory Server Certificates to the file system, eliminating the need to manually configure the server through iManager. Ensure that you select this option.

Table 3-1 shows the differences between the different versions of OES and their ability to support eDirectory certificates.

Table 3-1 *OES Versions and eDirectory Certificates*

OES Version	Supports eDirectory Certificates	Additional Information
OES 1	No	
OES 2	Yes	Only new server installations have the option to be automatically configured. Upgrades and post-installs do not have this option.
OES 2 SP1	Yes	The default is for all installations to have the option to be automatically configured.

NOTE: If use of eDirectory certificates is enabled while installing OES 2 (default), the install code creates a configuration for the SSL CertificateDNS object, and the certificates and keys are exported to the following files:

key file - `/etc/ssl/servercerts/serverkey.pem`

certificate file - `/etc/ssl/servercerts/servercert.pem`

3.6 Trusted Root Object Tasks

- Section 3.6.1, "Creating a Trusted Root Container," on page 57
- Section 3.6.2, "Creating a Trusted Root Object," on page 57

3.6.1 Creating a Trusted Root Container

This task is described in “Creating a Trusted Root Container” on page 25.

3.6.2 Creating a Trusted Root Object

This task is described in “Creating a Trusted Root Object” on page 25.

3.6.3 Viewing a Trusted Root Object's Properties

In addition to the eDirectory rights and properties that are viewable with any eDirectory object, you can also view properties specific to the Trusted Root object, including the issuer, the certificate status, and the validation period.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.
4. Browse to and click the Trusted Root object you want to view.
5. Click *OK*.
6. To view the certificate chain, click on the plus sign (+) in front of the certificate’s nickname to expand the view.
7. Click the nickname of the certificate to view its details.
8. Click *Cancel*.

3.6.4 Replacing a Trusted Root Certificate

This task allows you to replace a Trusted Root Certificate that is stored in the Trusted Root object. This task should be performed if the Trusted Root Certificate has expired.

You can replace a Trusted Root Certificate from the Trusted Root object's property page.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, “Entry Rights Needed to Perform Tasks,” on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.
4. Browse to and click the Trusted Root object you want to replace.
5. Click *OK*.
6. Select the certificate, then click *Replace*.

7 Browse for and select the new Trusted Root certificate.

8 Click *OK*.

3.6.5 Validating a Trusted Root Object

If you suspect a problem with a certificate or think that it might no longer be valid, you can easily validate the certificate by using iManager. Any certificate in the eDirectory tree can be validated, including certificates issued by external CAs.

The certificate validation process includes several checks of the data in the certificate as well as the data in the certificate chain. A certificate chain is composed of a root CA certificate and, optionally, the certificates of one or more intermediate CAs.

A result of Valid means that all certificates in the certificate chain were found to be valid. Certificates are considered valid if they pass a predefined set of criteria including whether the current time is within the validity period of the certificate, whether it has not been revoked, and whether it has been signed by a CA that is trusted. Only those certificates with a CRL distribution point extension or an OCSP AIA extension are checked for revocation.

A result of Invalid means that one or more certificates in the certificate chain were found to be invalid or their validity could not be determined. Additional information is provided in these cases about which certificate is considered invalid and why. Click *Help* for more information about the reason.

To validate a Trusted Root certificate:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.

4 Browse to and click the Trusted Root object you want to validate.

5 Click *OK*.

6 Select the certificate, then click *Validate*.

The status of the certificate is provided in the *Certificate Status field*. If the certificate is not valid, the reason is given.

NOTE: If the certificate in the object is not self-signed, its certificate chain must be in the Trusted Roots container in the Security container (CN=Trusted Roots.CN=Security) for the validation to succeed. Typically, the certificate chain consists of a single, root-level CA or it consists of an Intermediate CA and a root-level CA. The name of the Trusted Roots container must be Trusted Roots and each certificate in the chain must be stored in its own Trusted Root object. For instructions on how to create a Trusted Roots container and Trusted Root objects, see "Creating a Trusted Root Container" on page 57 and "Creating a Trusted Root Object" on page 57.

3.6.6 Revoking a Trusted Root Certificate

You might find it necessary to revoke a certificate if the key or the CA becomes compromised, if the certificate has been superseded by another certificate, if the certificate is removed from the CRL, cessation of operation, etc.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, click *Directory Administration* > *Modify Object*.

4 Browse to and click the Trusted Root object you want to modify.

5 Click *OK*.

6 Select the certificate, then click *Revoke*.

This starts the Revoke Certificate Wizard. Follow the prompts to revoke the certificate.

7 Click *Finish*.

3.7 Certificate Revocation List (CRL) Tasks

NetIQ Certificate Server provides a system for managing Certificate Revocation Lists (CRLs). This is an optional system, but it must be implemented if you want to be able to revoke certificates created by the Organizational CA.

A CRL is a published list of revoked certificates and the reason the certificates were revoked.

3.7.1 Creating a CRL Container Manually

During the Certificate Server installation, a CRL container is created if the user has the appropriate rights to create it. If not, the CRL container can be created manually by someone with the appropriate rights after the installation is completed.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.

If a CRL container already exists, you are brought to the Organizational CA's property page.

If no CRL container exists, this launches a wizard that creates a CRL container and a CRL Configuration object to go in the container.

4 Follow the wizard to completion.

3.7.2 Deleting a CRL Container

Deleting a CRL container is possible, but it is not recommended.

The general rule is to not delete a CRL container, CRL configuration object, CRL object, or CRL file until one issue date after the last certificate that contains a related distribution point has expired.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *Directory Administration* > *Delete Object*.
4. Browse for and select the CRL container you want to delete.
5. Click *OK* > *OK*.

3.7.3 Creating a CRL Configuration Object

A CRL Configuration object can be created in the CRL container. This is an object that contains the configuration information for the CRL objects that are available in the eDirectory tree. Normally, you have only one CRL Configuration object in your tree. You might need multiple CRL Configuration objects if you are creating or rolling over a new Organizational CA, but only one CRL Configuration object can be used to create new certificates.

The CRL Configuration object resides in the CRL container.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority* and then do one of the following:
   - If no CRL container exists, this launches a wizard that creates a CRL container and a CRL Configuration object to go in the container. Follow the wizard to completion.
   - If a CRL container exists, but no CRL Configuration object exists, this launches a wizard that creates a CRL Configuration object to go in the container. Follow the wizard to completion.
   - If a CRL container exists and a CRL Configuration object exists, you are brought to the Organizational CA's property page. Continue with Step 4.
4. Click the *CRL* tab.
5. Click *New*.
6. Type the name of the new CRL configuration object, then click *OK*.
7. Follow the wizard to completion.

3.7.4 Activating a CRL Configuration Object

Only one CRL Configuration object can be active in an eDirectory tree at one time. If you have more than one CRL Configuration object, you must choose which one to activate. By default, the first CRL Configuration object created is active.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.
4. Click the *CRL* tab.
5. Select a CRL Configuration object, then click *Make Active*.
6. Click *OK* or *Apply*.

3.7.5 Viewing and Modifying a CRL Configuration Object's Properties

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.
4. Click the *CRL* tab.
5. Click on the name of the CRL Configuration object you want to view or modify.
6. Click *OK* or *Apply*.

- "LDAP Mapping" on page 61
- "HTTP Distribution Point Location" on page 62

LDAP Mapping

The standard LDAP type for Certificate Revocation Lists limits the size of the CRL to 64 KB. To change this limitation, you must create the CRL directory entries with NetIQ-defined types. In order for the LDAP distribution points to be found, you must map the standard LDAP types to the NetIQ LDAP types by doing the following:

1. Launch iManager.
2. Log in to the eDirectory as an administrator with the appropriate rights.
3. On the *Roles and Tasks* menu, select *LDAP* > *LDAP Options*.
4. Click the *View LDAP Groups* tab, then select the LDAP group that needs to be mapped.

5 Click the *General* tab, select the Attribute Map page, and make the following changes:

5a The default mapping from Primary LDAP Attribute certificateRevocationList; binary (and secondary attribute certificateRevocationList) to the eDirectory attribute certificateAuthorityList should be changed to the eDirectory attribute ndspkiCertificateRevocationList (that is, change the eDirectory attribute from certificateAuthorityList to ndspkiCertificateRevocationList).

5b The default mapping from Primary LDAP Attribute authorityRevocationList;binary (secondary attribute authorityRevocationList) to the eDirectory attribute authorityRevocationList should be changed to the eDirectory attribute ndspkiAuthorityRevocationList (that is, change the eDirectory attribute from authorityRevocationList to ndspkiAuthorityRevocationList).

5c The default mapping from Primary LDAP Attribute deltaRevocationList;binary (secondary attribute deltaRevocationList) to the eDirectory attribute deltaRevocationList should be changed to the eDirectory attribute ndspkiDeltaRevocationList (i.e. change the eDirectory attribute from deltaRevocationList to ndspkiDeltaRevocationList).

6 Click OK.

7 On the *Roles and Tasks* menu, select *LDAP* > *LDAP Options*.

8 Click the *View LDAP Servers* tab, then select the server that hosts the LDAP distribution point.

9 Click the *General* tab, then select the Information page.

10 Click the refresh button.

This restarts the LDAP service, and it begins using the correct mapping for the CRL attributes.

For more information on LDAP management, see "Configuring LDAP Services for NetIQ eDirectory" in the *NetIQ eDirectory 8.8 SP8 Administration Guide*.

HTTP Distribution Point Location

When configuring Certificate Server to use an HTTP distribution point, it is important that you specify a location that is accessible to users wanting to validate certificates. If a user cannot locate a CRL for a certificate containing a distribution point, the certificate is considered invalid. The distribution point must be located in a directory that is available to the Web server specified by the HTTP address in the distribution point. If that directory is not on the same server that is hosting the Certificate Authority, the CRL must be moved manually, with a script, or created on a mounted directory.

3.7.6 Deleting a CRL Configuration Object

Deleting a CRL Configuration object is possible, but it is not recommended. When a CRL Configuration object is deleted, the server quits creating the CRL files. If a CRL file already exists in the location specified in the CRL object, certificate validation continues to use it until it expires. After it expires, all certificates that have a CRL distribution point that references that CRL file fail validation.

The general rule is to not delete a CRL container, CRL configuration object, CRL object, or CRL file until one issue date after the last certificate that contains a related distribution point has expired.

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93

3 On the *Roles and Tasks* menu, select *Directory Administration* > *Delete Object*.

4 Browse for and select the CRL Configuration object you want to delete.

5 Click *OK* > *OK*.

3.7.7 Creating a CRL Object

This task allows you to create a CRL object (cRLDistributionPoint) to store third-party CRLs in eDirectory. This object can be created in any container in the eDirectory tree. But as a general rule, NetIQ CRL objects reside in a CRL Configuration object and do not need to be created manually. A CRL object is automatically created for you when you create a CRL Configuration object.

The CRL object contains a CRL file, which contains the detailed CRL information. For a NetIQ CRL object, the CRL file is automatically created and updated whenever the server issues a new one. For other CRL objects, you must import a CRL file from a third-party CA.

NOTE: The term CRL Distribution Point is used in different ways. It is the eDirectory schema object name for the CRL object and it can be used in general terms as the point where the CRL information is published.

To create a CRL object:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Create CRL Object*.

4 Type a name for the object and provide the context where you want the object to reside.

5 Paste a copy of the CRL into the field or read it from a CRL file.

6 Click *Finish* to create the object.

3.7.8 Exporting a CRL File

You can export the CRL that is contained in the CRL Distribution Point object to a file.

To export a NetIQ CRL file:

1 Launch iManager.

2 Log in to the eDirectory tree as an administrator with the appropriate rights.

To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.

3 On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.

4 Click the *CRL* tab.

5 Click the name of the CRL Configuration object, then click *Details*.

6 Click *Export*.

7 Select an output format, then click *Next*.

8 To save the exported CRL to a file, click *Save*, then specify a location for the file.

9 Click *OK* > *OK*.

To export a third-party CRL file:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *Directory Administration* > *Modify Object*.
4. Browse for and select the CRL Configuration object, then click *OK*.
5. Click *Export*.
6. Select an output format, then click *Next*.
7. To save the exported CRL to a file, click *Save*, then specify a location for the file.
8. Click *OK* > *OK*.

3.7.9 Replacing a CRL File

You can replace a CRL file, but it is not recommended.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.
4. Click the *CRL* tab.
5. Click the name of the CRL Configuration object, then click *Details*.
6. Click *Replace*.
7. Click *OK* to continue.
8. Browse for and select the new CRL file.
9. Click *OK*.

If a CRL file does not exist on the CRL Configuration object, the *Import* button is displayed.

3.7.10 Viewing a CRL Object's Properties

To view a NetIQ CRL object's properties:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *NetIQ Certificate Server* > *Configure Certificate Authority*.
4. Click the *CRL* tab.
5. Click the name of the CRL Configuration object, then click *Details*.

   You can now view the CRL object's properties.
6. When you are finished viewing properties, click *OK* or *Apply*.

To view a third-party CRL object's properties:

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, select *Directory Administration* > *Modify Object*.
4. Browse to and click the CRL object you want to view, then click *OK*.
5. Click *Edit*.

   You can now view the CRL object's properties.
6. When you are finished viewing properties, click *OK* or *Apply*.

3.7.11 Deleting a CRL Object

If you delete a CRL object, it is re-created the next time the server generates the CRL file. If you delete a CRL object that you created using iManager and import it, then it is gone permanently and any certificates that reference it are considered invalid.

The general rule is to not delete a CRL container, CRL configuration object, CRL object, or CRL file until one issue date after the last certificate that contains a related distribution point has expired.

1. Launch iManager.
2. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropriate rights for this task, see Appendix B, "Entry Rights Needed to Perform Tasks," on page 93.
3. On the *Roles and Tasks* menu, click *Directory Administration* > *Delete Object*.
4. Browse to and click the CRL object you want to delete.
5. Click *OK* > *OK*.

3.8 eDirectory Tasks

3.8.1 Resolving Multiple Security Containers, Organizational CAs, KAP Containers, and W0 Objects

NetIQ Certificate Server can be installed on multiple servers in an eDirectory tree. However, for NetIQ Certificate Server to function properly, only one Security container, Organizational CA, KAP container, and W0 object should exist in the tree.

If you are installing NetIQ Certificate Server on multiple servers in an eDirectory tree, you must allow eDirectory to replicate between each installation of NetIQ Certificate Server. If you do not allow eDirectory to replicate, your installation to another server might not recognize that the tree already has a Security container, an Organizational CA, a KAP container, and a W0 object and might re-create these objects on another server in the same eDirectory tree.

The items below describe possible scenarios and how to resolve them.

- If you have two or more Security containers in the same eDirectory tree and each contains an Organizational CA, and a KAP container with a W0 object, do not issue any certificates. Contact Technical Support for help in resolving this.
- If you have one Security container that contains two KAP containers in the same eDirectory tree, do not issue any certificates. Contact Technical Support for help in resolving this.
- If you have one Security container that contains two Organizational CAs and one KAP container with a W0 object in the same eDirectory tree, delete every server and user certificate issued by both Organizational CAs. Then, delete both CAs and create a new Organizational CA. Issue new server and user certificates as needed.
- If you have two or more Security containers in the same eDirectory tree and each contains an Organizational CA, but only one contains a KAP container with a W0 object, delete every server and user certificate issued by all Organizational CAs. Delete all the Security containers without the KAP container and W0 object. If the remaining Security container is not named *Security*, rename it to *Security*. Issue new server and user certificates as needed.
- If you have two or more Security containers in the same eDirectory tree and only one contains an Organizational CA and a KAP container with a W0 object, delete all the Security containers without the KAP container and W0 object. If the remaining Security container is not named *Security*, rename it to *Security*.

3.8.2 Restoring or Re-creating a Security Container

If you delete the Security container, you cannot create an Organizational Certificate Authority until you have restored or re-created the security container.

To restore the security container, you must restore the eDirectory partition containing the Security container.

To re-create the Security container, use one of two methods:

- Using iManager, click *Directory Administration* > *Create Object*. Click *Tree's Security Container*, then click *OK*. The container name must be Security.
- Reinstall NetIQ Certificate Server on any server in the eDirectory tree.

3.8.3 Restoring or Re-creating KAP and W0

Do not delete the KAP or W0 objects. Doing so invalidates all previously created User certificates. If you delete one of these objects, go to the Novell Support Web site (http://support.novell.com/) and search for TID #3032354, "How to Restore or Recreate KAP and W0 Objects," for information on how to resolve this problem. You should not attempt further installations of NetIQ Certificate Server, Single Sign-on, NMAS, or eDirectory until the problems have been corrected.

3.9 Application Tasks

This section describes how to configure the GroupWise 6.x and 5.5 enhancement pack client, Groupwise 6.x, Outlook 98, Outlook 2000, and Netscape Messenger to use NetIQ certificates for secure e-mail. This section also describes how to configure other cryptography-enabled applications to use NetIQ certificates.

Some of the information in this section is dated but useful. For the latest information on using certificates with your cryptography-enabled applications, refer to the application's documentation.

The general process for enabling applications for secure e-mail is:

1. Export your Organizational CA's self-signed certificate (see "Exporting the Organizational CA's Self-Signed Certificate" on page 31), your user certificate, and the matching private key to a `.pfx` file (see "Exporting a User Certificate and Private Key" on page 48).
2. Import the `.pfx` file into your e-mail client. See Section 3.9.1, "Importing the User Certificate and Private Key into Your E-Mail Client," on page 67.
3. Configure your e-mail client to secure your e-mail. See Section 3.9.2, "Configuring Your E-Mail Client to Secure Your E-Mail," on page 68.

3.9.1 Importing the User Certificate and Private Key into Your E-Mail Client

Installing a user certificate and private key (a `.pfx` file) into Internet Explorer automatically makes it available for use by GroupWise and Microsoft Outlook. The reverse is also true. Installing the certificate and private key into either e-mail application automatically makes it available for use by the other e-mail application and by Internet Explorer.

Installing a user certificate and private key into Netscape automatically makes it available for use by Netscape Messenger. The reverse is also true.

Groupwise 6.x and GroupWise 5.5 Enhancement Pack Client

1. Launch GroupWise.
2. Click *Tools* > *Options*.
3. Double-click the *Certificates* icon.
4. Click *Import*.
5. Browse for and select or type the filename of your exported `.pfx` file.
6. Enter your password, then click *OK*.
7. Click *Set Security Level* if you want to change the default security level for your private key, then click *OK*.
8. To select a default certificate to use for sending signed e-mail, you can now either select the check box next to the certificate or select the certificate and click *Set as Default*.
9. Click *OK*.

Microsoft Outlook 98

1. Launch Outlook.
2. Click *Tools* > *Options*.
3. Click the *Security* tab.
4. Click *Import/Export Digital ID*.
5. Select the *Import existing Exchange or S/MIME Security Information* radio button.
6. For *Import File and Password*, type the filename and password of your exported `.pfx` file.

7 For *Keyset*, type a nickname. This can be any text.

8 Click *OK* to import the private key and certificate into Outlook 98.

Microsoft Outlook 2000

This procedure applies to Outlook 2000 with Microsoft Internet Explorer version 5.

1 Launch Outlook.

2 Click *Tools* > *Options*.

3 Click the *Security* tab.

4 Click *Import/Export Digital ID*.

5 Select the *Import existing Exchange or S/MIME Security Information* radio button.

6 For *Import File and Password*, type the filename and password of your exported `.pfx` file.

7 For *Digital ID Name*, type a nickname. This can be any text.

8 If you are prompted to add the Organizational CA certificate to the Root Store, click *Yes*.

Netscape Messenger 4.*x*

1 Launch Netscape Messenger.

2 Click the *New Msg* icon.

3 Double-click the *Security* icon on the Navigation toolbar.

4 Click *Certificates* > *Yours*.

5 Click *Import a Certificate*. If you password-protected the Communicator Certificate database, enter the password.

6 Type or browse for and select the filename of the exported `.pfx` file.

7 Type the password you used to protect the `.pfx` file.

8 Click *OK*.

3.9.2 Configuring Your E-Mail Client to Secure Your E-Mail

The following describes how to configure e-mail clients for secure e-mail.

- "GroupWise 6.x Client" on page 68
- "GroupWise 5.5 Enhancement Pack Client" on page 69
- "Microsoft Outlook" on page 70
- "Netscape Messenger" on page 70

GroupWise 6.x Client

You must have imported at least one certificate and private key (`.pfx` file) into GroupWise or Internet Explorer in order to make use of signed e-mail. You must also have a certificate available for each recipient that you want to send encrypted email to.

1 Launch GroupWise.

2 Click *Tools* > *Options*.

3 Click the *Security* tab.

4. Click the *Send Options* tab.
5. To enable signing as the default for all outgoing email, select the check box next to *Sign Digitally*. To enable encryption as the default for all outgoing e-mail, select the check box next to *Encrypt for Recipients*.
6. Click *OK*.
7. Double-click the *Certificates* icon.
8. Select the certificate that you want to use for signing, encryption, or both, then click the *Set As Default* button.

   If the certificate can be used for both signing and encryption, it is the default certificate used for both signing and encryption. If you have two certificates, one that can only be used for signing and one that can only be used for encryption, the former should be set as the default for signing and the latter as the default for encryption.

From an item view (send mail, post message, task, reminder note, etc.), you can change the default security options for this particular item by selecting *File* > *Properties* and clicking the *Security* tab. From here you can change the signing and encryption options.

From an item view (send mail, post message, task, reminder note, etc.), you can also toggle the selection of either signing or encryption for this particular item by clicking the *Encrypt* or *Digitally Sign* icons at the top of the view.

GroupWise 5.5 Enhancement Pack Client

You must have imported at least one certificate and key into GroupWise in order to make use of signed e-mail. You must also have a certificate available for each recipient that you want to send encrypted e-mail to.

1. Launch GroupWise.
2. Click *Tools* > *Options*.
3. Double-click the *Security* icon.
4. Click the *Send Options* tab.
5. To enable signing as the default for all outgoing e-mail, select the check box next to *Sign Digitally Using*. You can then select a different certificate to use from the Certificate drop-down list below this option.
6. To enable encryption as the default for all outgoing e-mail, select the check box next to *Encrypt for Recipient Using*, then select the encryption method from the *Method* drop-down list below this option. The available encryption methods depend on the security service provider you have selected.
7. To select a different security service provider, select a provider from the *Name* drop-down list, then click *OK*.

From an item view (send mail, post message, task, reminder note, etc.), you can change the default security options for this particular item by selecting *File* > *Properties* and clicking the *Security* tab. From here you can change the signing and encryption options.

From an item view (send mail, post message, task, reminder note, etc.), you can also toggle the selection of either signing or encryption for this particular item by clicking the *Encrypt* or *Digitally Sign* icons at the top of the view.

Microsoft Outlook

1. Launch Outlook.
2. Click *Tools* > *Options*.
3. Click the *Security* tab.
4. Click either *Setup Secure E-Mail* or *Change Settings*, depending on whether you have previously entered security settings.
5. Select *S/MIME* for the *Secure Message Format*.
6. Click the *Choose* button on the *Signing Certificate* line.
7. Select the certificate that you will use for digitally signing e-mail that you send to others, then click *OK*.
8. Click the *Choose* button on the *Encryption Certificate* line.
9. Select the certificate that others will use for encrypting e-mail that they send to you, then click OK.
10. Select the *Send These Certificates with Signed Message* check box, then click *OK*.
11. Select the combination of options you prefer in the *Secure E-Mail* section, then click *OK*.

Netscape Messenger

1. Launch Netscape Messenger.
2. Click the *New Msg* icon.
3. Click the *Security* icon.
4. Click *Messenger*.
5. Select the certificate you will use for digitally signing your e-mail that you send to others under the *Certificate Signed and Encrypted Messages* heading.

   You can select other options as desired on this page. Refer to the Netscape help topics for further information on these options and their purposes.

3.9.3 Configuring Your Browser or E-Mail Client to Accept Certificates

In order to accept signed e-mail from another person or to create an SSL connection to a server on the Internet with your browser, you must trust the CA that signed the user or server's certificates. If you do not, your application might present you with an error. Some applications provide a warning with the ability to accept or reject the user or server certificate whose CA isn't yet known to the application.

Server and user certificates signed by a company's Organizational CA always generate such warnings and errors. This is because the Organizational CA is not listed as a trusted CA in your application. The warnings and errors can be prevented by installing the self-signed certificate of the Organizational CA into your application.

Installing the Organizational CA into Internet Explorer automatically adds it as a trusted CA to Microsoft Outlook and GroupWise. Installing the Organizational CA certificate into Netscape automatically adds it as a trusted CA to Netscape Messenger.

To accept the Organizational CA as a trusted CA in your application, first export the Organizational CA's self-signed certificate as described in "Exporting the Organizational CA's Self-Signed Certificate" on page 31. Then import it into your browser according to the directions below.

NOTE: The following Internet browsers only recognize certificates that have been exported in `.der` or a `.crt` format. Although `.b64` is an optional export format, it is not recognized by these Internet browsers.

Microsoft Internet Explorer Version 5 and 6

If you are using Microsoft Internet Explorer version 5, complete the following to import the Organizational CA's certificate:

1. Launch Microsoft Internet Explorer.
2. Click *File* > *Open*.
3. Type or browse for and select the filename of the exported Organizational CA's self-signed certificate, then click *OK*.

   This opens the Certificate dialog box.
4. Select *Install Certificate*.

   This opens the Certificate Manager Import Wizard.
5. Click *Next*.
6. Select the area where you want to store the certificate, click *Next*, click *Finish*, then click *Yes*.

Netscape Navigator

If you have installed either Microsoft Internet Explorer 5.*x* or NT 4 Service Pack 4 or later on your workstation, you must complete the following steps to import the Organizational CA's self-signed certificate into Netscape Navigator. This is necessary because the Microsoft products intercept opening trusted root files with a `.crt` or `.der` extension.

1. Run the `x509.reg` file to install the X.509 extension. On an NT\2000 server, this file is located in the *`drive_letter`*`:novell\nds directory`.
2. Rename the Organizational CA's self-signed certificate file with an X.509 extension.
3. Launch Netscape Navigator.
4. Click *File* > *Open Page*.
5. Enter or browse for and select the filename of the self-signed certificate with the X.509 extension.
6. Click *Open*.

   The New Certificate Authority dialog box should appear. If it doesn't, you have not correctly installed the .x509 extension, or you have not correctly renamed the self-signed certificate.
7. Follow the wizard. Make sure that the *Accept this Certificate Authority for Certifying E-Mail Users* check box is selected.
8. Click *Next* until the dialog box to enter a short name for this Certificate Authority appears.
9. Click *Finish*.

If you have not installed either Microsoft Internet Explorer 5.*x* or NT 4 Service Pack 4 or later, you must complete the following steps to import the Organizational CA's certificate into Netscape Navigator:

1. Launch Netscape Navigator.
2. Select *File* > *Open Page*.
3. Enter or browse for and select the filename of the self-signed certificate you previously exported.
4. Click *Open*.
5. Follow the wizard. Make sure the *Accept this Certificate Authority for Certifying E-Mail Users* check box is selected.
6. Click *Next* until the dialog box to enter a short name for this Certificate Authority appears.
7. Click *Finish*.

3.9.4 Configuring Microsoft Internet Explorer (IE) for SSL with NetIQ Certificates

To configure IE to use NetIQ certificates for SSL, you must first install your self-signed Organizational CA certificate in your IE browser, as described in "Configuring Your Browser or E-Mail Client to Accept Certificates" on page 70. Otherwise, any attempt to use IE to connect to a server that is using NetIQ certificates for SSL only displays an error.

This configuration is not strictly necessary for the Netscape browser, which will present a dialog box for you to accept or reject a server certificate whose CA isn't yet known to the browser.

3.9.5 Configuring Microsoft IIS for Client Authentication with NetIQ Certificates

To perform client authentication to IIS with NetIQ user certificates, your self-signed Organizational CA certificate must first be installed in IIS as a trusted root. Use Microsoft Internet Explorer (IE) version 4 or later to install your Organizational CA certificate on the IIS computer as described in the IIS online documentation.

However, the IISCA program described in the IIS documentation does not work on Windows NT with Service Pack 4 or later. In this case, when you use IE to install the certificate and the Certificate Manager Import Wizard has started, perform the following to complete the process correctly:

1. Select *Place All Certificates into the Following Store*.
2. Click *Browse* to open the *Select Certificate Store* dialog box.
3. Select the *Show Physical Stores* check box.
4. Expand *Trusted Root Certification Authorities* and select *Local Computer*.
5. Click *OK* > *Next* to open the *Completing the Certificate Manager Import Wizard* summary page.
6. Verify that the summary displays *Certificate Store Selected by User* and *Trusted Root Certification Authorities/Local Computer*.
7. Click *Finish*.
8. Stop and restart the IIS services after installing your Organizational CA certificate.

For further information, refer to Microsoft Knowledgebase articles Q218445 and Q216339.

3.9.6 Requesting a Server Certificate for Microsoft IIS

When using the IIS management tools to create an SSL key pair and certificate signing request (CSR), select *Put the Request in a File that You Will Send to an Authority in the Create New Key Wizard*.

Then edit the IIS CSR to delete all text that precedes the line:

```
----- BEGIN NEW CERTIFICATE REQUEST -----
```

This line must be the first line in the CSR input to the NetIQ Certificate Server. Refer to the IIS online documentation for further instructions on installing the resulting server certificate and configuring IIS for SSL.

You can then use your Organizational CA to issue a server certificate from the IIS CSR as described in "Issuing a Public Key Certificate" on page 30.

3.10 PKI Health Check

NetIQ Certificate Server incorporates a process that maintains the health and integrity of the Certificate Server components. This process is called the PKI Health Check and it runs when:

- The server reboots.
- eDirectory comes up.
- DSRepair finishes running.

When PKI Health Check runs, it performs the following tasks:

Table 3-2 *PKI Health Check Tasks*

Task	Function
Verifies the server's link to the SAS Service object	This task checks to see if there is a link from the server object to a SAS:Service object. If the link exists, the task makes sure that the object is named correctly and is in the same context as the server. If the link does not exist, the task checks to see if a correctly named object exists in the same context as the server. If such an object exists, the task creates a link from the server to the object.
Verifies the SAS Service object	This task checks to see if a SAS:Service object exists. If it does not exist, the task creates one and creates a link from the server object to the new object. Then, the task checks to see if the SAS:Service object has the necessary eDirectory rights. If it does not, the task attempts to give the SAS:Service object the rights it needs.
Verifies the links to the KMOs	This task reads the list of Server Certificate objects (or KMOs) that are linked to the SAS:Service object. It checks whether the KMOs are all named correctly and attempts to fix their names if they are not. The task also checks whether the KMOs are all in the same context as the server object and attempts to move them to the correct context if they are not.

Task	Function
Checks the Server Certificates (KMOs)	This task reads all the names of KMOs that are in the same container as the server object and puts them in a list. The task then performs the following for each KMO in the list: ◆ Attempts to populate the NDSPKI:Not Before and NDSPKI:Not After attributes with the validity dates of the certificate. ◆ Checks whether Public has the Read right to the Host Server attribute. ◆ Checks the link from the KMO to a server that is a back link. If the back link is for a different server, it ignores the KMO and removes it from its list. ◆ Reads the private key and attempts to unwrap it.
Reverifies the links to the KMOs	This task reads the list of Server Certificate objects (or KMOs) that are linked to the SAS:Service object. It compares each KMO in this list to the list created in Checks the Server Certificates (KMOs). Using the checks from Checks the Server Certificates (KMOs), the task determines if there are any problems with the linked certificates and it unlinks them if the KMO is unusable. The task also determines if there are any unlinked KMOs that are usable by this server and it links them, if they exist.
Creates default certificates	This task determines if Server Self-Provisioning is enabled at the Organizational CA object. If Server Self-Provisioning is not enabled, this step is skipped. If Server Self-Provisioning is enabled, then the task calls the NPKICreateDefaultCertificates() API. This API creates or replaces the SSL CertificateDNS certificate if: ◆ The certificate does not exist. ◆ The certificate is not expired or about to expire. ◆ The certificate's subject name does not match the default IP and DNS address configured for the server. **NOTE:** eDirectory 8.8 SP8 does not automatically create SSL CertificateIP. SSL Certificate DNS contains all the IPs listed in the Subject Alternative Name. In addition, this API acquires all of the IP and DNS addresses configured for the server and it creates and/or replaces a certificate for each one, such as IP AG *ip address* or IP DNS *dns name* if: ◆ The certificates do not exist. ◆ The certificates are expired or about to expire.

Task	Function
Synchronizes certificates for external services	This task reads the configuration from the SAS:Service object. For each configured entry, the task acquires the certificates and private key from the specified KMO object. If the specified directory does not exist, the task attempts to create it. The task then unwraps the private key and converts it to the specified raw-key format. The task compares any existing private key and certificate files to the ones from the specified KMO. If the keys and certificates are not the same, the task makes a backup of the existing private key and certificate files and then it overwrites them with the private key and certificates. The keys are written out in a `PEM` format.
Exports the eDirectory CA certificate to the file system	The way in which this task is accomplished depends on the operating system you are running. ◆ **Windows:** Checks if the `SSCert.der` and `SSCert.pem` files in the PKI working directory contain the same certificate as the Organization CA certificate in eDirectory. It attempts to replace the files if they are not the same. The default PKI working directory is `c:\Novell\NDS\DIBFiles\CertServ\` ◆ **Linux/AIX/Solaris (Not OES Linux):** Checks if the `SSCert.der` and `SSCert.pem` files in the eDirectory data directory contain the same certificate as the Organizational CA certificate in eDirectory. It attempts to replace the files if they are not the same. The default eDirectory data directory is `/var/opt/novell/eDirectory/data` ◆ **OES Linux:** Checks if the `/etc/opt/novell/certs/SSCert.der` and `/etc/opt/novell/certs/SSCert.pem` files contain the same certificate as the Organizational CA certificate in eDirectory. If the certificates are not the same, the task attempts to replace the files by adding the Organizational CA certificate to the `/etc/ssl/certs` directory and then running the `c_rehash` program. Before replacing the files, however, the task creates backups of any existing certificates.

4 Troubleshooting

This section provides troubleshooting information for known issues.

If you do not find a solution to your issue here, check the Readme file that accompanied the software as well as the Novell Support information database (http://support.novell.com).

4.1 Using PKIDiag

PKIDiag is a utility designed to diagnose and fix Certificate Server objects. PKIDiag can be used to do the following:

- Rename or move server-related objects so that they conform to the correct naming and containment scheme if a server has been moved.
- Create required objects if they do not exist.
- Grant the necessary rights between objects.
- Link objects if they are not linked.
- Create the SSL CertificateIP and the SSL CertificateDNS certificates if they do not exist.
- Fix the SSL CertificateIP and the SSL CertificateDNS certificates if either has an incorrect name, is out of date, or is close to expiring.

To load PKIDiag, at a server prompt type

```
Load PKIDiag
```

To see a list of command line options, at a server prompt type

```
Load PKIDiag /?
```

The functionality of PKIDiag is used by two other processes, the server auto health check and the Create Default Certificate task in iManager.

The server auto health check is run whenever a server is restarted or whenever DSREPAIR is run. Create Default Certificate is a process you use to replace the default certificates created when you install Certificate Server. See Section 3.2.2, “Creating Default Server Certificate Objects,” on page 37 for more information.

See TID #3640106 (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3640106&sliceId=SAL_Public&dialogID=2494290&stateId=1%200%202492620) for more information about PKIDiag and how it can be used.

4.2 Installation Issues

4.2.1 File Data Conflict During Installation

If you receive a message indicating that a newer file exists from the previous installation, you should select to always overwrite the newer file.

4.2.2 Incomplete List of Servers

The list of servers shown during the installation might not list servers that are configured to use only IP. You can install NetIQ Certificate Server on a server whose name is not listed by typing the name of the server in the text box.

4.2.3 Error Creating SAS Service Object During Install

When installing NetIQ Certificate Server, you might encounter an error stating that the Security Domain key server could not be contacted. The first server in your network that you install NetIQ Single Sign-on, NMAS, or NetIQ Certificate Server on is set up to be the Security Domain key server.

All subsequent servers that are installed with any of these products contact the Security Domain key server during their installation process. If the Security Domain key server cannot be contacted, the installation fails and a message is displayed indicating that the SAS service object could not be created.

To avoid the SAS service creation error message:

- Make sure the Security Domain key server machine is up and running nicisdi.nlm.

- Use a common protocol (IP and/or IPX™) between the Security Domain key server and all other servers that are installed with NetIQ Single Sign-on, NMAS, and NetIQ Certificate Server.
- Make sure the network connection between the Security Domain key server and the server being installed is active.

You can determine which server is the Security Domain key server by running iManager. Open the properties page for the W0 object. This object is located in the KAP container, which is inside the Security Container. Click the *Other* tab. Click NDSPKI:SD Key Server DN. The value displayed is the distinguished name of the Security Domain key server.

4.2.4 NISP:GET_PDB_PRODUCT:Returned a BTRIEVE error:4

If you receive this error one or more times during the installation, ignore it and continue with the installation.

4.2.5 Failures During Installation

If the installation fails during the creation of the Organizational CA or the server certificate, or during the exportation of the trusted root certificate, the installation doesn't need to be repeated. The software has been successfully installed at this point. You can use iManager to create an Organizational CA and server certificates and export the trusted root.

4.2.6 Installation Fails with a -1443 Error

If a NetIQ Certificate Server installation fails during installation and you receive a -1443 error message, this means that the Security Domain key server and the server that you are installing Certificate Server on are not communicating properly. If the server cannot get a copy of the Security Domain key, the installation fails.

A likely reason is that the server that Certificate Server is being installed to fragments of the (NetWare Core Protocol) NCP™ extensions, and the fragments are not being reassembled correctly by the Security Domain key server.

One solution to this problem is to increase the MTU of both servers to greater than 576 (the default minimum size).

To increase the MTU on a server:

1. Enter `LOAD MONITOR !h` from the command line of the server.
2. Select *Server Parameters* > click *Communications*.
3. Select *Maximum Interface MTU*, then set this value to something higher than 576.

4.2.7 PKI Plug-In Encounters Error When Installed on iManager 2.7.6 Patch1 and Lower Versions

To work around this issue, create a libntls.so.8 symbolic link pointing to libntls.so as follows:

```
ln -sf /var/opt/novell/iManager/nps/WEB-INF/bin/linux/libntls.so
/var/opt/novell/iManager/nps/WEB-INF/bin/linux/libntls.so.8
```

4.2.8 IP Auto Generated Certificate Is Not Created on SLES 11 64-Bit Platform

Consider a scenario where eDirectory 8.8 SP8 has both IPv4 and IPv6 configured and only one of the them (for example, IPv4) has an entry in the /etc/hosts file, and the other interface is accessible from a remote machine. If you configure eDirectory to listen on both the IPs, the IP AG certificate is generated only for the IP that is listed in the /etc/hosts file. In this example, it is generated for IPv4.

4.2.9 IP Auto Generated IPv6 Certificate is Not Created When the Length of the Certificate Object RDN Exceeds the Maximum Limit

While installing eDirectory 8.8 SP8, which is listening on both IP v4 and IPv6 addresses, IP AG <IPv6> certificate (KMO) is not created.

This occurs when the length of the RDN of the certificate object exceeds the maximum limit of 64 characters. To handle this, a compressed format of IPv6 address is used so that even if the length exceeds the maximum limit, the address is split to accommodate the request. The address is split from the third colon (from the reverse order) in the address.

For example, if the IPv6 address is 2508:f0g0:1003:0061:0000:0000:0000:0002, then the truncated address is 0000:0000:0002. This ensures that the host is identifiable even after the address is truncated.

4.2.10 HTTP Server Associates With the IP AG Certificate When the Default Server Certificates are Recreated for a Server where CA is not Hosted

Use iManager to manually change the default association.

Log in to iManager > Modify > Select the http server object > Select the httpKeyMaterialObject attribute, then change the HTTP server object association to SSL CertificateDNS.

4.3 User Certificate Issues

4.3.1 Waiting for Servers to Synchronize

Occasionally, after the user certificate has been created, the client is unable to refresh the view to include the new certificate. A dialog box is shown with the message “Waiting for servers to synchronize.” At this point, the user certificate has been created but the servers involved in the creation have not yet synchronized. You can close the dialog box without impacting the creation of the user's certificate.

4.3.2 Error Reusing Certificate Nicknames

If an error occurs during user certificate creation, try using a different nickname for the certificate. The nickname that was specified might not be available for reuse.

4.3.3 -1426 Error Exporting a User's Private Key

All servers with replicas of the partition in which the User object resides should have the same level of cryptography (U.S./Worldwide NICI or Import Restricted NICI). If they do not, an error of -1426 might appear when exporting the user's private key if the key size is too large.

To export the user's private key after a -1426 error has occurred, you must either upgrade the cryptography on the servers with replicas of the partition or remove the replica from those servers that have exportable cryptography.

4.3.4 Workstation Cryptography Strength

If U.S/Worldwide (high-grade) cryptography is loaded on your NetWare® server, you have the option to create user certificates with key sizes of 512, 768, 1024, 2048 and 4096 bits. However, any key size larger than 512 bits cannot be used with GroupWise® 5.5, Outlook 98, Outlook 2000, or Netscape Messenger unless you also have high-grade cryptography installed on the client workstation.

4.4 Server Certificate Issues

4.4.1 Server Uses Expired SSL CertificateIP Certificate

eDirectory 8.8 SP8 does not support the SSL CertificateIP certificate. If you upgrade to eDirectory 8.8 SP8 from a previous version, the SSL CertificateIP certificate remains associated with the server. When the certificates in your environment expire, the SSL CertificateIP certificate does not renew automatically.

Any time after you upgrade to eDirectory 8.8 SP8, you can start using the SSL CertificateDNS certificate instead of the SSL CertificateIP certificate.

4.4.2 External CAs

Some third-party CAs such as VeriSign use an intermediate CA to sign server certificates. In order to import these certificates into a Server Certificate object, the server certificate as well as the Intermediate CA and the trusted root certificate must be in a single PKCS #7 formatted file (`.P7B`). If your CA cannot provide you with such a file, you can create one yourself by following these steps on a client machine with Internet Explorer 5.5 or later installed.

1. Import the server certificate into Internet Explorer. You can do this by double-clicking on the file or by selecting *File* > *Open* and selecting the filename.
2. If the external CA's certificate is not already listed as a trusted CA in Internet Explorer, import the Intermediate CAs as well as the root level CA in the same manner.

3. In Internet Explorer, select *Tools* > *Internet Options*. Select the *Content* tab, then select the *Certificates* button.
4. On the *Personal* tab, find the server certificate. Select it and click *Export*.
5. Accept the defaults in the wizard until you get to the Export File Format page, then select the Cryptographic Message Syntax Standard - PKCS #7 Certificates (.p7b) format.
6. Continue with the wizard.

   The PKCS #7 file can now be imported into the Server Certificate object.

4.4.3 Moving a Server

If a Server object is moved, the LDAP objects, SAS service object, and Server Certificate objects (Key Material Objects) for that server should also be moved. But the server auto health check will move the objects for you the next time you restart the server.

4.4.4 DNS Support

If DNS is configured for the server, the default subject name for a server certificate will be:

.CN=<*Server's DNS Name*>.O=<*Tree Name*>

Otherwise, the default subject name is the fully distinguished name of the server. You can modify the default subject name by selecting Custom during the certificate creation process.

4.4.5 Removing a Server from eDirectory

When removing a server from eDirectory™ and then reinstalling it into the same context with the same name, a successful reinstallation occurs only if the SAS Service object representing the removed server is also deleted, if it existed.

The process should go like this:

1. Determine if the default certificates need to be backed up. If so, back them up.
2. Delete the default certificates.
3. Delete the SAS object.

For example, for a server named MYSERVER, a SAS object named SAS Service - MYSERVER could exist in the same container as the server. This SAS object must be manually deleted (using iManager) after the server is removed from the tree, but before the server is reinstalled into the tree.

If the server is the Organizational CA or the SD Key server, you must complete some additional steps. These steps are documented in TID #3623407 (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3623407&sliceId=SAL_Public&dialogID=24943325&stateId=1%200%202492660).

The default server certificates created for the server should also be removed so that they are re-created when the server is reinserted.

These certificates are SSL Certificate IP - MYSERVER and SSL Certificate DNS - MYSERVER. You should be careful when deleting these certificates. If data has been encrypted by using either of these certificates, the data must be retrieved before the certificates are deleted.

4.4.6 Step-Up Cryptography, Server-Gated Cryptography, or Global Certificates

Some external Certificate Authorities provide certificates that enable 40-bit or 56-bit Web browser clients to use 128-bit cryptography when communicating with a server configured with their certificates.

These certificates are sometimes referred to as global certificates or server-gated cryptography certificates. The capability can be referred to as step-up cryptography.

These certificates can be used successfully for LDAP and Web Server connections only if the Web browser has 128-bit cryptography. Web browsers with 40-bit or 56-bit cryptography experience unrecoverable SSL errors when communicating with servers configured with these certificates.

If Web browsers with 40-bit or 56-bit cryptography must communicate with your server, you must request a different type of certificate from your external CA.

4.4.7 Subject Name Limitations for CAs

Server certificates with an @ character in their subject names might cause SSL connections to fail. Contact Technical Support for a resolution of the problem.

4.5 Validation Issues

4.5.1 Certificate Validation Speed

The certificate validation process includes several checks of the data in the certificate as well as the data in the certificate chain. A certificate chain is composed of a Root CA certificate and, optionally, the certificates of one or more intermediate CAs.

Validating the information in a certificate and its associated certificate chain is not a time-intensive process. However, there are occasions where the validation might take longer:

- If the certificate was signed by an external CA and one or more of the certificates has a CRL distribution point extension.

  In order to validate the certificate, the CRL for each applicable certificate in the chain must be retrieved. The CRL must then be examined to determine whether or not the certificate has been revoked.

  If the CRLs are large or if the server operating the CRL distribution point is busy, it might take some time to validate a certificate. The time required can be decreased by doing one or more of the following:

  - Upgrade the speed of the connection being used to check the revocation status of the certificate.
  - Contact your CA provider.
- If one or more of the certificates has an OCSP AIA extension. If the OCSP responder is busy, it might take a significant amount of time to validate.
- If you are validating a user certificate.

For server certificates, the entire certificate chain is stored along with the server certificate in the Key Material object. Therefore, when a server certificate is validated, the client can get all of the certificates necessary by simply reading one object. User certificates, however, are stored differently. Only the user certificate itself is stored in the User object. Thus, the client must retrieve the certificate chain from other objects stored in the Security container in order to validate the user certificate.

In order to validate a user certificate signed by the Organizational CA, the client must read the Organizational CA's object in order to retrieve the CA's certificate. In order to validate a user certificate signed by an external CA, the client must read the Trusted Roots container in the Security container in order to compose a certificate chain that matches the user certificate. In the latter case, an Administrator must have already imported the certificates of the external CAs into the Trusted Roots container in order for the validation of the User certificate to succeed.

The time required to validate a user certificate can be decreased by removing expired certificates that are no longer trusted from the Trusted Roots container.

4.5.2 Validating Certificates after Deleting the Organizational CA

If you delete the Organizational CA (other than during a backup and restore procedure), you should export the self-signed certificate and create a new trusted root in the trusted roots container. If you don't, you will experience the following behavior when validating these certificates:

- User certificates signed by the deleted CA are invalid. This is because the certificate of the CA that signed the user certificate could not be found in the Organizational CA object or in the Trusted Roots container. If you want those user certificates to remain valid, you must add the previous CA's self-signed certificate to the Trusted Roots container.
- Server certificates signed by the deleted CA continue to be valid. This is because the CA's certificate is stored in the Key Material object along with the server certificate.

  If you deleted the Organizational CA because the key had been compromised or because of some security breach, you should immediately revoke all user and server certificates that were signed by the CA. If you cannot revoke them, you should delete them and create new certificates to take their place. You should also tell all users who might have imported your Organizational CA's certificate into their browsers to delete the certificate.

4.6 Miscellaneous Issues

4.6.1 -1497 Errors

If you receive a -1497 error while the `pki.nlm` is loading, or as a result of performing certificate management, the probable cause is that NICI has not been installed correctly or has become corrupted.

To resolve the problem, reinstall NICI and retry the operation. If that does not solve the problem, call Technical Support.

4.6.2 Renaming the Security Container

You cannot rename the security container.

A Public Key Cryptography Basics

This sections describes the basics of Public Key Cryptography.

A.1 Overview

The content of most Internet communications, such as Web page browsing or public chat forums, can be monitored by anyone equipped to do so. The content of other data transmissions, such as the exchange of credit card information for online purchases, needs to be kept private.

Public key cryptography is a widely used method for keeping data transmissions private and secure on the Internet. Specifically, public key cryptography is the system of using digital codes called “keys” to authenticate senders of messages and to encrypt message content.

A.2 Secure Transmissions

Data transmissions are private and secure when two things happen:

- **Authentication:** The data receiver knows that the data sender is exactly who or what it claims to be.
- **Encryption:** The data sent is encrypted so that it can be read only by the intended receiver.

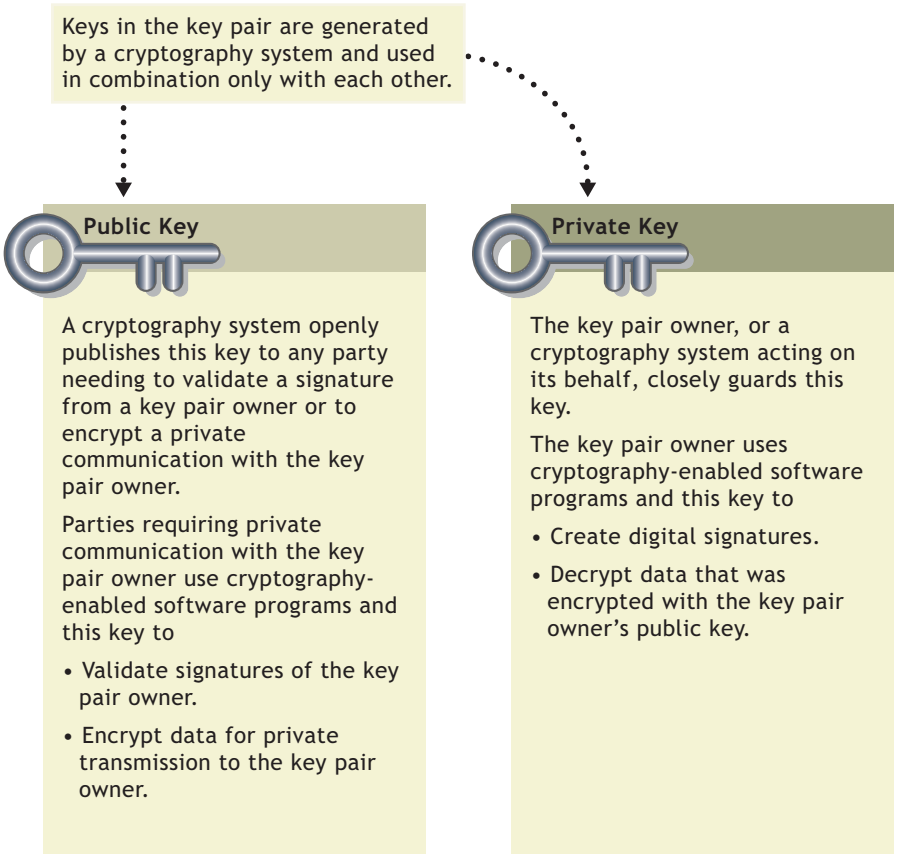
A.3 Key Pairs

Authentication and encryption are both provided through the use of mathematically related pairs of digital codes or “keys.” One key in each pair is publicly distributed; the other is kept strictly private.

Each data transmitter, whether it is a person, a software program, or some other entity such as a bank or business, is issued a key pair by a public key cryptography system.

The basic principles and functions of each key in the key pair are summarized in the following illustration:

Figure A-1 *Basic Key Pair Description*



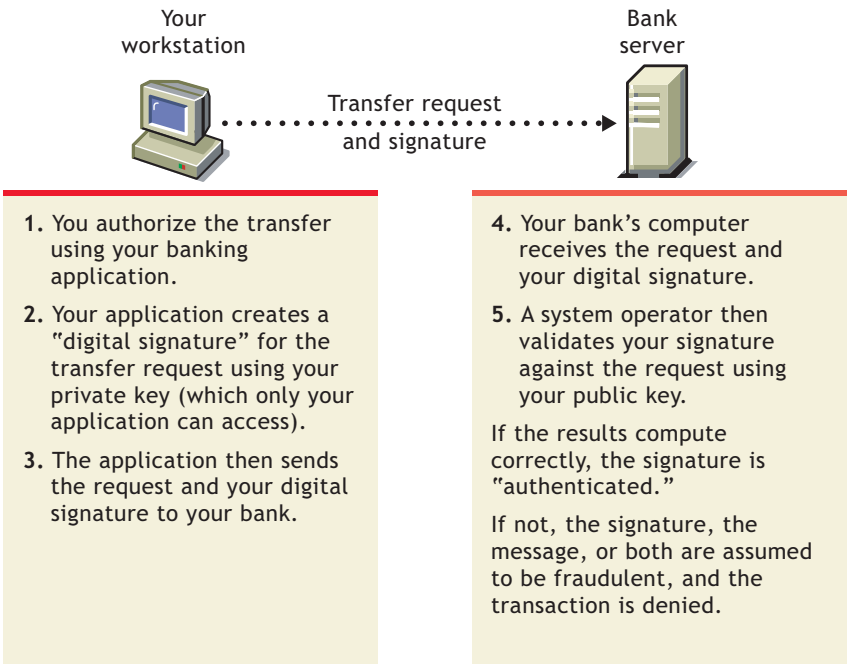
- Section A.3.1, "Key Pairs and Authentication," on page 86
- Section A.3.2, "Key Pairs and Encryption," on page 87

A.3.1 Key Pairs and Authentication

Authentication means that the data receiver knows that the data sender is exactly who or what it claims to be.

Suppose that you want to authorize your bank to transfer funds from your account to another account. The bank needs proof that the message came from you and that it has not been altered during transit. The following illustrates the process that your online transaction would follow, using public key cryptography.

Figure A-2 *Public Key Process*



For information about digital signatures and their verification, see "Digital Signatures" on page 89.

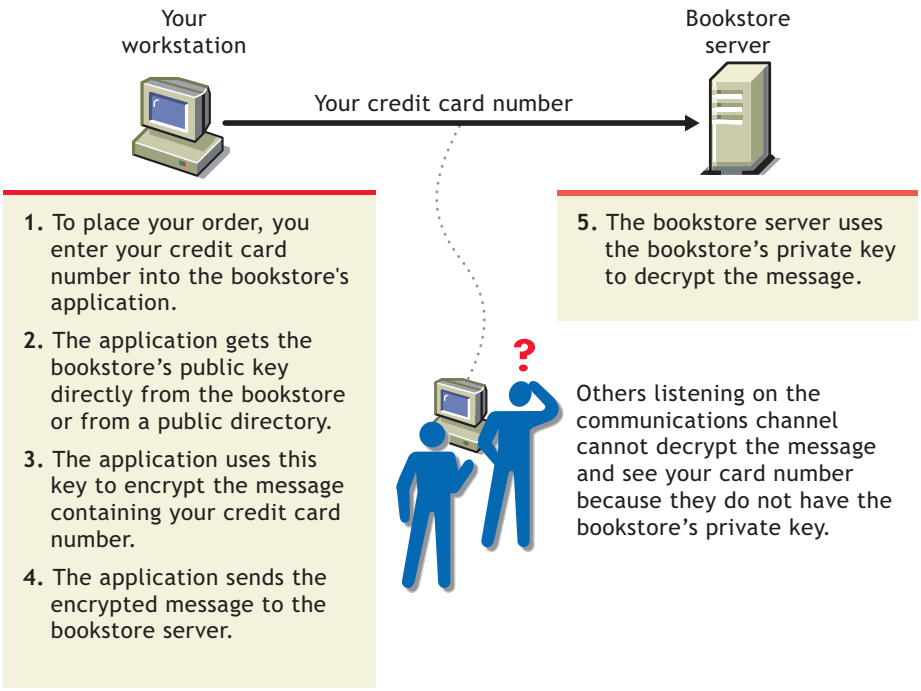
A.3.2 Key Pairs and Encryption

Encryption means that the data can be read only by the intended receiver.

Suppose you want to order a book from an Internet vendor and you need to use your credit card to pay for it. You don't want your credit card number read by anyone other than the intended recipient.

The encryption process in the following illustration provides the mechanisms through which your credit card number can be safely transmitted.

Figure A-3 *Encryption Process*



A.4 Establishing Trust

If a sender and receiver know and trust each other, they can simply exchange public keys and establish secure data transmission, including authentication and encryption. To do this, they would use each other's public keys and their own private keys.

Under normal circumstances, however, parties needing secure data transmissions have no foundation for trusting the identity of each other. Each needs a third party, whom they both trust, to provide proof of their identity.

- Section A.4.1, “Certificate Authorities,” on page 88
- Section A.4.2, “Digital Signatures,” on page 89
- Section A.4.3, “Certificate Chain,” on page 90
- Section A.4.4, “Trusted Roots,” on page 91

A.4.1 Certificate Authorities

A party needing to prove its identity in a public key cryptography environment enlists the services of a trusted third party known as a certificate authority.

The primary purpose of the certificate authority is to verify that a party is who or what it claims to be, and then to issue a public key certificate for that party to use. The public key certificate verifies that the public key contained in the certificate belongs to the party named in the certificate.

Figure A-4 *Certificate Request*

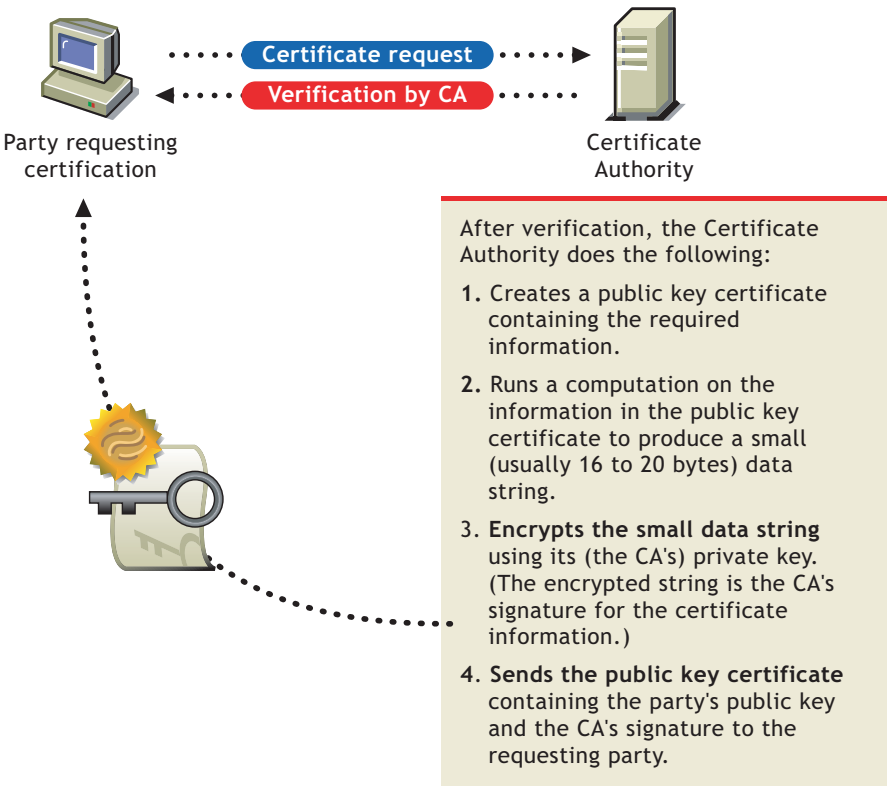
After the identity of the requesting party has been established to the satisfaction of the certificate authority, the certificate authority issues an electronic "certificate" and applies its digital signature.

A.4.2 Digital Signatures

Just as a personal signature applied to a paper document indicates the authenticity of the document, a digital signature indicates the authenticity of electronic data.

To create a digital signature, the software used to create the signature links the data being signed with the private key of the signer. The following illustration shows the process that a CA follows to create its digital signature for a public key certificate.

Figure A-5 *Digital Signature*



A digital signature is uniquely linked to the signer and the data. No one else can duplicate the signature because no one else has the signer's private key. In addition, the signer cannot deny having signed the data. This is known as *non-repudiation*.

When a certificate authority signs a public key certificate, it guarantees that it has verified the identify of the public key owner according to the certificate authority's established and published policies.

After signed data (such as a public key certificate) is received, software verifies data authenticity by applying the same computation to the data that the signing software used originally. If the data is unaltered, both computations produce identical results. It can then be safely assumed that neither the data nor the signature was modified in transit.

A.4.3 Certificate Chain

A certificate chain is an ordered list of certificates. The certificates are ordered such that the server or user certificate is first, followed by the certificate of its CA.

CAs can either sign their own certificates (that is, they are self-signed) or they can be signed by another certificate authority. If they are self-signed, they are typically called root CAs. If they are not self-signed, they are typically called subordinate CAs or intermediate CAs.

If a user or server certificate was signed by a CA with a self-signed certificate, the certificate chain is composed of exactly two certificates: the end entity certificate and the root CA.

If a user or server certificate was signed by an intermediate CA, then the certificate chain is longer. The first two elements are still the end entity certificate, followed by the certificate of the intermediate CA. But the intermediate CA's certificate are then followed by the certificate of its CA. This listing then continues until the last certificate in the list is for a root CA. Thus, a certificate chain can be infinitely long. In practice, however, most certificate chains have only two or three certificates.

A.4.4 Trusted Roots

In order to validate a digital signature, you must trust at least one of the certificates in the user or server's certificate chain. You can directly trust the certificate of the user or server, or you can choose to trust any other certificate in the chain. Typically, the certificate that is trusted is the root CA's certificate.

Most application software that can use certificates already has a list of trusted certificates installed. These certificates are for root CAs and, hence, are called "trusted roots."Typically these CAs are commercial CAs. If you choose, you can add additional CAs to this list or remove CAs from the list.

B Entry Rights Needed to Perform Tasks

The following list provides the specific entry rights an administrator needs to manage NetIQ Certificate Server tasks within an eDirectory tree. These rights are the minimum entry rights needed.

This list should also be helpful to the administrator who wants to grant rights to another user to manage part or all of company's certificate authority and certificate management needs.

Table B-1 *Administrator Entry Rights*

Tasks	Entry Rights Needed
Install NetIQ Certificate Server	For the first installation to an eDirectory tree: ◆ Supervisor at the [Root] of the tree For subsequent installations: ◆ Supervisor to the W0 object ◆ Rights needed to create a Server Certificate object If a user doesn't have the rights to create a Server Certificate object, the installation finishes, but the Server Certificate objects need to be created manually by someone with the appropriate rights and applications that use these certificates need to be manually configured.
Creating an Organizational CA	◆ Supervisor on the Security container
Viewing the Organizational CA's properties and certificates	◆ Browse on the Organizational CA's object
Exporting the Organizational CA's certificate(s)	◆ Browse on the Organizational CA's object
Issuing a public key certificate	◆ Read to the NDSPKI:Private Key on the Organizational CA's object However, if the object trying to issue the public key certificate is an NCP server, then the rights needed are: ◆ Write to the NDSPKI:Private Key on the Organizational CA's object
Backing up and restoring an Organizational CA	◆ Supervisor on the Organizational CA's object

Tasks	Entry Rights Needed
Moving the Organizational CA to a different server	◆ Supervisor on the Organizational CA's object
Validating the Organizational CA's Certificates	◆ Browse on the Organizational CA's object
Replacing the Organizational CA	◆ Supervisor on the Organizational CA's object
Deleting the Organizational CA	◆ Delete on the Organizational CA's object
Creating Server Certificate objects	◆ Supervisor on the server's container ◆ Read to the attribute NDSPKI:Private Key on the Organizational CA's object (only if using the Organizational CA) However, if the object trying to issue the public key certificate is an NCP server, then the rights needed are: ◆ Supervisor on the server’s container ◆ Write to the NDSPKI:Private Key on the Organizational CA’s object
Importing a public key certificate into a Server Certificate object	◆ Write to the attribute NDSPKI:Public Key Certificate on the Server Certificate object ◆ Write to the attribute NDSPKI:Certificate Chain on the Server Certificate Object
Deleting a Server Certificate object	◆ Delete on the Server Certificate object
Exporting a Trusted Root or Public Key Certificate from a Server Certificate object	◆ Browse on the Server Certificate object
Viewing the Server Certificate object's properties and certificates	◆ Browse on the Server Certificate object
Backing up and restoring a Server Certificate object	◆ Supervisor on the server object that owns the Server Certificate object to back-up ◆ Create on the Server object's container to restore.
Validating Server Certificates	◆ Browse on the Server Certificate object
Revoking Server Certificates	◆ Read to the CA Private Key or Delete on the Server Certificate object or Supervisor on the Host Server (that is, the NCP™ Server object)
Replacing a server certificate's keying material	◆ Write to the attribute NDSPKI:PrivateKey on the Server Certificate object

Tasks	Entry Rights Needed
Creating user certificates	◆ Read to the attribute NDSPKI:Private Key on the Organizational CA object ◆ Read and Write to the attribute NDSPKI:userCertificateInfo on the User object ◆ Read and Write to the attribute SAS:SecretStore on the User object ◆ Read and Write to the attribute userCertificate on the User object However, if the object trying to issue the public key certificate is an NCP server, then the rights needed are: ◆ Write to the NDSPKI:Private Key on the Organizational CA's object ◆ Read and Write to the attribute NDSPKI:userCertificateInfo on the User object ◆ Read and Write to the attribute SAS:SecretStore on the User object ◆ Read and Write to the attribute userCertificate on the User object
Importing a public key certificate into a User object	◆ Read and Write on the attribute NDSPKI:userCertificateInfo on the User object ◆ Read and Write to the attribute NDSPKI:userCertificate on the User object
Viewing a user certificate's properties	◆ Browse on the User object
Exporting a user certificate	◆ Browse on the User object
Exporting a user's private key and certificate	◆ You must be logged in as the user
Deleting a user certificate and private key	◆ Read and Write to NDSPKI:userCertificateInfo ◆ Read and Write to userCertificate
Validating User Certificates	◆ Browse on the User object
Revoking User Certificates	◆ Read to the CA Private Key or Delete on the User Object or be logged-in as the User and Write to the userCertificate attribute
Creating a Trusted Root Container	◆ Create on the Security container
Creating a Trusted Root object	◆ Create on the Trusted Root Container in which the Trusted Root object will reside
Viewing a Trusted Root object's properties	◆ Browse on the Trusted Root object

Tasks	Entry Rights Needed
Replacing a trusted root certificate	◆ Read and Write to NDSPKI:Not After on the Trusted Root object ◆ Read and Write to NDSPKI:Not Before on the Trusted Root object ◆ Read and Write to NDSPKI:Subject Name on the Trusted Root object ◆ Read and Write to NDSPKI:Trusted Root Certificate on the Trusted Root object
Validating a trusted root certificate	◆ Browse on the Trusted Root object
Revoking a trusted root certificate	◆ Read to the CA Private Key or Delete on the Trusted Root object
Deleting a Trusted Root object	◆ Delete on the Trusted Root object
Creating a CRL Container	◆ Supervisor on the Security container ◆ Write to the attribute ndspkiCRLContainerDN on the Organizational CA's object
Deleting a CRL Container	◆ Delete on the CRL container
Creating a CRL Configuration object	◆ Supervisor on the CRL container
Activating a CRL Configuration object	◆ Write to the attribute ndspkiCRLConfigurationDNList on the Organizational CA's object
Viewing and/or Modifying a CRL Configuration object's Properties	Modifying: ◆ Supervisor on the CRL Configuration object or ◆ Write to the attribute being modified on the CRL Configuration object Viewing: ◆ Browse on the CRL Configuration object
Deleting a CRL Configuration object	◆ Delete on the CRL Configuration object
Creating a CRL object	◆ Supervisor of the CRL Configuration object
Exporting a CRL file	◆ Read from the attribute certificateRevocationList
Replacing a CRL file	◆ Browse on the CRL object
Viewing a CRL object's properties	◆ Browse to the attribute certificateRevocationList
Deleting a CRL object	◆ Delete on the CRL Distribution Point
Creating a Security container	◆ Create at the root of the eDirectory tree

Tasks	Entry Rights Needed
Creating a SAS service object	◆ Supervisor on the object's container ◆ Write to the attribute SAS:Service DN on the server that the object is being created for.