Novell Certificate Server™

2.5.2

www.novell.com

ADMINISTRATION GUIDE

July 1, 2003

Novell®

Legal Notices

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

Novell Certificate Server 2.5.2 Administration Guide
July 1, 2003

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

ConsoleOne is a registered trademark of Novell, Inc., in the United States and other countries.
eDirectory is a trademark of Novell, Inc.
GroupWise is a registered trademark of Novell, Inc., in the United States and other countries.
IPX is a trademark of Novell, Inc.
NDS is a registered trademark of Novell, Inc., in the United States and other countries.
NetWare is a registered trademark of Novell, Inc., in the United States and other countries.
NCP is a trademark of Novell, Inc.
NDS is a trademark of Novell, Inc.
NLM is a trademark of Novell, Inc.
NMAS is a trademark of Novell, Inc.
Novell is a registered trademark of Novell, Inc., in the United States and other countries.
Novell Authorized Reseller is a service mark of Novell, Inc.
Novell Modular Authentication Services is a trademake of Novell, Inc.
Novell Certificate Server is a trademark of Novell, Inc.
Novell Client is a trademark of Novell, Inc.
Novell Directory Services is a trademark of Novell, Inc.
Novell SecretStore is a registered trademark of Novell, Inc., in the United States and other countries.

Third-Party Trademarks

All third-party trademarks are the property of their respective owners.

Contents

About This Guide 9

1 Overview 11

Product Components . . . 12
- Novell Certificate Server . . . 12
- Novell International Cryptographic Infrastructure . . . 15
- For Additional Information . . . 15

2 Setting Up Novell Certificate Server 17

Deciding Which Type of Certificate Authority to Use . . . 17
- Benefits of Using an Organizational Certificate Authority Provided with Novell Certificate Server . . . 17
- Benefits of Using an External Certificate Authority . . . 18

Creating an Organizational Certificate Authority Object . . . 18
Creating Server Certificate Objects . . . 19
- Hints for Creating Server Certificates . . . 20

Configuring Cryptography-Enabled Applications . . . 20
Additional Components to Set Up . . . 20
- Create a User Certificate . . . 20
- Creating a Trusted Root Container . . . 21
- Creating Trusted Root Objects . . . 21
- Creating a SAS Service Object . . . 22

3 Managing Novell Certificate Server 23

Certificate Authority Tasks . . . 25
- Creating an Organizational Certificate Authority Object . . . 25
- Issuing a Public Key Certificate . . . 25
- Viewing the Organizational CA's Properties . . . 26
- Viewing an Organizational CA's Public Key Certificate Properties . . . 26
- Viewing the CA's Self-Signed Certificate Properties . . . 27
- Exporting the Organizational CA's Self-Signed Certificate . . . 27
- Backing Up an Organizational CA . . . 28
- Restoring an Organizational CA . . . 29
- Moving the Organizational CA to a Different Server . . . 29
- Validating the Organizational CA's Certificates . . . 30
- Replacing the Organizational CA . . . 30
- Deleting the Organizational CA . . . 31

Server Certificate Object Tasks . . . 32
- Creating Server Certificate Objects . . . 32
- Importing a Public Key Certificate into a Server Certificate Object . . . 32
- Exporting a Trusted Root or Public Key Certificate . . . 34
- Deleting a Server Certificate Object . . . 34
- Viewing a Server Certificate Object's Properties . . . 35
- Viewing a Server Certificate Object's Public Key Certificate Properties . . . 35
- Viewing a Server Certificate Object's Trusted Root Certificate Properties . . . 35
- Backing Up a Server Certificate Object . . . 36
- Restoring a Server Certificate Object . . . 37

Server Certificate Objects and Clustering . . . 38
Validating a Server Certificate . . . 38
Moving a Server Certificate Object to a Different Server . . . 39
Replacing a Server Certificate Object's Keying Material . . . 39
User Certificate Tasks . . . 40
Creating User Certificates . . . 40
Creating User Certificates in Bulk . . . 40
Importing a Public Key Certificate into a User Object . . . 41
Viewing a User Certificate's Properties . . . 41
Exporting a User Certificate Using ConsoleOne . . . 42
Exporting a User Certificate and Private Key Using ConsoleOne . . . 42
Deleting a User Certificate and Private Key . . . 43
Validating a User Certificate . . . 43
Trusted Root Object Tasks . . . 44
Creating a Trusted Root Container . . . 44
Creating a Trusted Root Object . . . 44
Viewing a Trusted Root Object's Properties . . . 44
Replacing a Trusted Root Certificate . . . 45
Validating a Trusted Root Object . . . 45
Certificate Revocation List (CRL) Tasks . . . 46
Creating a Certificate Revocation List (CRL) Object . . . 46
Importing a Third-Party CRL . . . 47
Exporting a Third-Party CRL . . . 47
Replacing a Third-Party CRL . . . 47
Viewing a Third-Party CRL's Properties . . . 48
eDirectory Tasks . . . 48
Merging Two Trees that Have Security Containers . . . 48
Resolving Multiple Security Containers, Organizational CAs, KAP Containers, and W0 Objects . . . 52
Restoring or Re-creating a Security Container . . . 53
Restoring or Re-creating KAP and W0 . . . 53
Application Tasks . . . 53
Importing the User Certificate and Private Key into Your E-Mail Client . . . 53
Configuring Your E-Mail Client to Secure Your E-Mail . . . 55
Configuring Your Browser or E-Mail Client to Accept Certificates . . . 57
Configuring Microsoft Internet Explorer (IE) for SSL with Novell Certificates . . . 58
Configuring Microsoft IIS for Client Authentication with Novell Certificates . . . 59
Requesting a Server Certificate for Microsoft IIS . . . 59

4 Troubleshooting 61

Installation Issues . . . 61
File Data Conflict During Installation . . . 61
Incomplete List of Servers . . . 61
Error Creating SAS Service Object During Install . . . 61
NISP:GET_PDB_PRODUCT:Returned a BTRIEVE error:4 . . . 62
Failures During Installation . . . 62
Installation Fails with a -1443 Error . . . 62
User Certificate Issues . . . 62
Waiting for Servers to Synchronize . . . 62
Error Reusing Certificate Nicknames . . . 62
-1426 Error Exporting a User's Private Key . . . 63
Workstation Cryptography Strength . . . 63
Server Certificate Issues . . . 63
External CAs . . . 63
Moving a Server . . . 63
DNS Support . . . 63
Deletion of the SAS Object . . . 64

Removing a Server from a Tree . . . 64
Step-Up Cryptography, Server-Gated Cryptography, or Global Certificates . . . 64
Subject Name Limitations for CAs . . . 64
ConsoleOne Issues . . . 64
ConsoleOne on a Server . . . 64
Refresh After Update . . . 64
Organizational CA Issues . . . 65
Path Length . . . 65
Moving the Organizational CA Object . . . 65
Validation Issues . . . 65
Certificate Validation Speed . . . 65
Validating Certificates after Deleting the Organizational CA . . . 66
Miscellaneous Issues . . . 66
-1497 Errors . . . 66
Renaming the Security Container . . . 66
Certificate Encodings . . . 66

A Public Key Cryptography Basics 69

Overview . . . 69
Secure Transmissions . . . 69
Key Pairs . . . 69
Key Pairs and Authentication . . . 70
Key Pairs and Encryption . . . 71
Establishing Trust . . . 72
Certificate Authorities . . . 72
Digital Signatures . . . 72
Certificate Chain . . . 73
Trusted Roots . . . 74

B Entry Rights Needed to Perform Tasks 75

About This Guide

Novell® 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.

This book describes the functionality of Novell 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.

- Chapter 1, “Overview,” on page 11
- Chapter 2, “Setting Up Novell Certificate Server,” on page 17
- Chapter 3, “Managing Novell Certificate Server,” on page 23
- Chapter 4, “Troubleshooting,” on page 61
- Appendix A, “Public Key Cryptography Basics,” on page 69
- Appendix B, “Entry Rights Needed to Perform Tasks,” on page 75

Documentation Updates

For the most recent version of the *Novell Certificate Server 2.5.2 Administration Guide*, see the Novell Certificate Server Web site (http://www.novell.com/documentation/lg/crt252/index.html).

Documentation Conventions

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

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

1 Overview

Novell® Certificate Server™ provides public key cryptography services that are natively integrated into Novell 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 69.

Public key cryptography presents unique challenges to network administrators. Novell 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 Crytography 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 using standard eDirectory backup utilities.

- Allows central administration of certificates using ConsoleOne®. You can also perform some administration tasks using Novell iManager.

  ConsoleOne snap-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. The Novell iManager plug-in also allows you to perform some administration tasks.

- Allows users to manage their own certificates

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

- Supports popular e-mail clients and browsers

  Novell Certificate Server allows you to create and manage user certificates for securing e-mail. Novell 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 both Netscape Navigator* and Microsoft Internet Explorer.

Product Components

Novell Certificate Server

Novell Certificate Server consists of the PKI server component, a snap-in module to ConsoleOne, and a plug-in module to Novell iManager. ConsoleOne and Novell iManager are the administration points for Novell Certificate Server.

Novell 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.

Novell 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 Novell Certificate Server can be used in installations throughout the world.

After installing Novell Certificate Server, you will manage it using:

- ConsoleOne running on a client. (Novell Certificate Server cannot be managed using ConsoleOne running on a NetWare server console.)
- Novell iManager.

Through ConsoleOne and Novell iManager, you can perform the following tasks:

- Create an Organizational certificate authority for your organization (ConsoleOne and Novell iManager)

  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.

- Create a Server Certificate object for each cryptography-enabled application (ConsoleOne and Novell iManager)

  During the installation, you can elect to create a Server Certificate object. 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 created 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 filed when viewing the Server Certificate object in ConsoleOne.

NOTE: 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.

- Create a user certificate (ConsoleOne and Novell iManager)

  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 created by other CAs can only be imported after having been created.

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

- Create a Trusted Root Container (ConsoleOne)

  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.

  You must create the Trusted Root Container in the Security Container.

- Create a Trusted Root object (ConsoleOne and Novell iManager)

  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 will 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.

- Create certificates for external users and servers (ConsoleOne and Novell iManager)

  The CA administrator can use the Organizational CA to sign certificates for users and servers that reside in other trees. Such certificates are requested using a 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 using the Issue Certificate tool in ConsoleOne or Novell 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 (ConsoleOne)

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

  Certificates are considered valid if they pass a pre-defined 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 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.

- Export private keys and certificates (ConsoleOne)

  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.

- Import private keys and certificates (ConsoleOne)

  You can choose to import a key rather than create a new one at the time a Server 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 or to move the Organizational CA from one server to another.

  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 a SAS service object (Novell iManager)

  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. Likewise, 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 will not be able to create new server certificates.

  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 will add 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 ConsoleOne or Novell iManager to create a new one. If there are server certificates that belong to this server, you will need to link them up to the SAS service object manually by using the Other tab from ConsoleOne.

Novell International Cryptographic Infrastructure

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

NICI must be installed on the server in order for Novell Certificate Server to work properly. NICI does not ship with Novell Certificate Server. The proper version of NICI might be provided when Novell Certificate Server is bundled with another product, such as NetWare or eDirectory, or you might need to download a newer version of NICI from Novell's Software Download Web site (http://www.novell.com/download). When Novell Certificate Server is bundled with other products, you might be required to install NICI manually, or NICI might be automatically installed. Refer to the product's installation guide for more information.

For Additional Information

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

For instructions on setting up Novell Certificate Server, see Chapter 2, "Setting Up Novell Certificate Server," on page 17.

For information about administering Novell Certificate Server, see Chapter 3, "Managing Novell Certificate Server," on page 23.

For the latest online documentation for this and other Novell products, see the Product Documentation Web site (http://www.novell.com/documentation).

For additional information about this and other Novell security products and technologies, see the Novell Security Web site (http://www.novell.com/security).

2 Setting Up Novell Certificate Server

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

- "Deciding Which Type of Certificate Authority to Use" on page 17
- "Creating an Organizational Certificate Authority Object" on page 18
- "Creating Server Certificate Objects" on page 19
- "Configuring Cryptography-Enabled Applications" on page 20

Deciding Which Type of Certificate Authority to Use

Novell 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 only be signed by the Organizational CA.

During the Server Certificate object creation process, you will be 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.

An external Certificate Authority is managed by a third party outside of the eDirectory tree. An example of an external 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.

Benefits of Using an Organizational Certificate Authority Provided with Novell Certificate Server

- **Compatibility.** The Organizational Certificate Authority is compatible with Novell applications such as LDAP services, Portal Services, and the Novonyx Web Server. Certificates issued by the Organizational CA are X.509 v3 compliant and can also be used by third-party applications.
- **Cost 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 Novell eDirectory™ without relying on any external services. In addition, Novell Certificate Server is compatible with a wide range of 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.

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.

Creating an Organizational Certificate Authority Object

By default, the Novell Certificate Server installation process will create the Organizational Certificate Authority (CA) for you. You will be 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 using ConsoleOne® or Novell iManager. Also, if you delete the Organizational CA, you will need to re-create it.

IMPORTANT: During the creation process, you will be prompted to name the Organizational Certificate Authority object and to choose a server on which the Certificate Authority service will run.

Select a server that is physically secure, that will be available when needed to perform signing operations, that runs a protocol that is compatible with the other servers in your organization (for example, IP, IPX™, IP/IPX), and that only runs software that you trust. It is important that your server meet these conditions, because the Organizational Certificate Authority object is the centerpiece of your PKI system and if the server that contains the object is compromised, your entire PKI system could be compromised as well.

To create the Organizational Certificate Authority object using ConsoleOne:

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

   To view the appropriate rights for this task, see "Creating an Organizational CA" on page 75.
2. Start ConsoleOne.
3. Expand the eDirectory tree where you want to create the Organizational Certificate Authority.

   This reveals the Security container object.
4. Right-click the Security container object, then click New > Object.
5. From the list box in the New Object dialog box, double-click NDSPKI:Certificate Authority.

   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.

To create the Organizational Certificate Authority object using Novell iManager:

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

To view the appropriate rights for this task, see “Creating an Organizational CA” on page 75.

3 From the Roles and Tasks menu, click PKI Certificate Management > Create Certificate Authority.

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.

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

Creating Server Certificate Objects

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.

The Novell Certificate Server installation process can create a Server Certificate object for you. You will be prompted to specify a Server Certificate object name. When you click Finish, the Server Certificate object is created with the default parameters and placed in the container where the target server resides.

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

To create additional Server Certificate objects using ConsoleOne:

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

To view the appropriate rights for this task, see “Creating Server Certificate objects” on page 75.

2 Start ConsoleOne.

3 Right-click the container object that contains the server that will run your cryptography-enabled applications, then click New > Object.

4 From the list box in the New Objects dialog box, double-click NDSPKI:Key Material.

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.

To create additional Server Certificate objects using Novell iManager:

1 Launch Novell iManager.

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

To view the appropriate rights for this task, see “Creating Server Certificate objects” on page 75.

3 From the Roles and Tasks menu, click PKI Certificate Management > 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.

Hints for Creating Server Certificates

During the Server Certificate object creation process, you will be 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 Novell Certificate Server, and its name is based on the key pair name that you choose.

If you choose the Custom creation method, you will also be 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 "Deciding Which Type of Certificate Authority to Use" on page 17.

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 will generate a certificate signing request that you will need to submit to the external Certificate Authority.

After the certificate is signed and returned to you, you will need to install it into the Server Certificate object, along with the trusted root for the external Certificate Authority. For specific information on any of the wizard pages, click Help.

After you have created the Server Certificate object, you can configure your applications to use it. (See "Configuring Cryptography-Enabled Applications" on page 20.) Keys are referenced in the application's configuration by the key pair name that you entered when you created the Server Certificate object.

Configuring Cryptography-Enabled Applications

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

See "Application Tasks" on page 53 for specific instructions on configuring GroupWise® 5.5 or later, Outlook 98, Outlook 2000, and Netscape Messenger.

Additional Components to Set Up

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

Create a User Certificate

To create a user certificate using ConsoleOne:

1. Log in to the eDirectory tree as an administrator with the appropriate rights. To view the appropriate rights for this task, see "Creating user certificates" on page 76.

2. Start ConsoleOne.
3. Double-click the User object that will host the user certificate.
4. Click the Security tab > Certificates.
5. Click Create.

   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.

To create a user certificate using Novell iManager:

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

   To view the appropriate rights for this task, see "Creating user certificates" on page 76.
3. From the Roles and Tasks menu, click PKI Certificate Management > 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.

Creating a Trusted Root Container

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

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

   To view the appropriate rights for this task, see"Creating a Trusted Root Container" on page 76.
2. Start ConsoleOne.
3. Right-click the container you want to create the Trusted Root container in, then click New > Object.
4. From the list box in the New Object dialog box, double-click NDSPKI:Trusted Root.

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

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. Novell 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.

Creating Trusted Root Objects

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

To create Trusted Root objects using ConsoleOne:

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

   To view the appropriate rights for this task, see "Creating a Trusted Root object" on page 76.
2. Start ConsoleOne.
3. Open the Security container.
4. Right-click the Trusted Root Container object, then click New > Object.

5 From the list box in the New Object dialog box, double-click NDSPKI:Trusted Root Object.

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.

To create Trusted Root objects using Novell iManager:

1 Launch Novell iManager.

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

To view the appropriate rights for this task, see "Creating a Trusted Root object" on page 76.

3 From the Roles and Tasks menu, click PKI Certificate Management > 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).

Creating a SAS Service Object

To create a SAS Service Object using Novell iManager:

1 Launch Novell iManager.

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

To view the appropriate rights for this task, see "Creating a SAS service object" on page 77.

3 From the Roles and Tasks menu, click PKI Certificate Management > 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 Novell Certificate Server

As a system administrator, you will need to perform several tasks to maintain the public key cryptography services provided through Novell® Certificate Server™. Most of these tasks are performed within ConsoleOne®. Some tasks are also performed using Novell iManager, and some tasks are performed using the Novell Certificate Console utility. This chapter provides a brief overview and specific information on completing each task.

Certificate Authority tasks:

- "Creating an Organizational Certificate Authority Object" on page 18
- "Issuing a Public Key Certificate" on page 25
- "Viewing the Organizational CA's Properties" on page 26
- "Viewing an Organizational CA's Public Key Certificate Properties" on page 26
- "Viewing the CA's Self-Signed Certificate Properties" on page 27
- "Exporting the Organizational CA's Self-Signed Certificate" on page 27
- "Backing Up an Organizational CA" on page 28
- "Restoring an Organizational CA" on page 29
- "Moving the Organizational CA to a Different Server" on page 29
- "Validating the Organizational CA's Certificates" on page 30
- "Replacing the Organizational CA" on page 30
- "Deleting the Organizational CA" on page 31

Server Certificate object tasks:

- "Creating Server Certificate Objects" on page 19
- "Importing a Public Key Certificate into a Server Certificate Object" on page 32
- "Exporting a Trusted Root or Public Key Certificate" on page 34
- "Deleting a Server Certificate Object" on page 34
- "Viewing a Server Certificate Object's Properties" on page 35
- "Viewing a Server Certificate Object's Public Key Certificate Properties" on page 35
- "Viewing a Server Certificate Object's Trusted Root Certificate Properties" on page 35
- "Backing Up a Server Certificate Object" on page 36
- "Restoring a Server Certificate Object" on page 37
- "Server Certificate Objects and Clustering" on page 38
- "Validating a Server Certificate" on page 38

- “Moving a Server Certificate Object to a Different Server” on page 39
- “Replacing a Server Certificate Object’s Keying Material” on page 39

User Certificate tasks:

- “Create a User Certificate” on page 20
- “Creating User Certificates in Bulk” on page 40
- “Importing a Public Key Certificate into a User Object” on page 41
- “Viewing a User Certificate’s Properties” on page 41
- “Exporting a User Certificate Using ConsoleOne” on page 42
- “Exporting a User Certificate and Private Key Using ConsoleOne” on page 42
- “Deleting a User Certificate and Private Key” on page 43
- “Validating a User Certificate” on page 43

Trusted Root object tasks:

- “Creating a Trusted Root Container” on page 21
- “Creating Trusted Root Objects” on page 21
- “Viewing a Trusted Root Object’s Properties” on page 44
- “Replacing a Trusted Root Certificate” on page 45
- “Validating a Trusted Root Object” on page 45

Certificate Revocation List (CRL) Tasks:

- “Creating a Certificate Revocation List (CRL) Object” on page 46
- “Importing a Third-Party CRL” on page 47
- “Exporting a Third-Party CRL” on page 47
- “Replacing a Third-Party CRL” on page 47
- “Viewing a Third-Party CRL’s Properties” on page 48

Novell eDirectory® tasks:

- “Merging Two Trees that Have Security Containers” on page 48
- “Resolving Multiple Security Containers, Organizational CAs, KAP Containers, and W0 Objects” on page 52
- “Restoring or Re-creating a Security Container” on page 53
- “Restoring or Re-creating KAP and W0” on page 53

Application tasks:

- “Importing the User Certificate and Private Key into Your E-Mail Client” on page 53
- “Configuring Your E-Mail Client to Secure Your E-Mail” on page 55
- “Configuring Your Browser or E-Mail Client to Accept Certificates” on page 57
- “Configuring Microsoft Internet Explorer (IE) for SSL with Novell Certificates” on page 58
- “Configuring Microsoft IIS for Client Authentication with Novell Certificates” on page 59
- “Requesting a Server Certificate for Microsoft IIS” on page 59

Certificate Authority Tasks

Creating an Organizational Certificate Authority Object

This task is described in Chapter 2. See "Creating an Organizational Certificate Authority Object" on page 18.

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 using ConsoleOne:

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

   To view the appropriate rights for this task, see "Issuing a public key certificate" on page 75.
2. Start ConsoleOne.
3. Click a container object.
4. On the menu bar, click Tools > Issue Certificate.
5. Paste a Certificate Signing Request (CSR) into the dialog box, or use the Browse button to locate a CSR file and open it in the dialog box.
6. Click Next.
7. Select the Organizational Certificate Authority (CA) to sign the certificate, then click Next.
8. Specify how the key is to be used, then click Next.
9. Specify the subject name, the validity period, and the effective and expiration dates, then click Next.
10. 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.

To issue a public key certificate using Novell iManager:

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

   To view the appropriate rights for this task, see "Issuing a public key certificate" on page 75.
3. From the Roles and Tasks menu, click PKI Certificate Management > Issue Certificate.
4. Use the Browse button to locate a CSR file, open the file, then click Next.

5 Specify how the key is to be used, then click Next.

6 Specify the subject name, the validity period, and the effective and expiration dates, then click Next.

7 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.

Viewing the Organizational CA's Properties

Besides 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.

These properties provide you with the information that you need to perform any task related to this object.

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

To view the appropriate rights for this task, see "Viewing the Organizational CA's properties and certificates" on page 75.

2 Start ConsoleOne.

3 Double-click the Organizational CA object.

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

4 Click the tabs that you want to view.

Viewing an Organizational CA's Public Key Certificate Properties

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

To view the appropriate rights for this task, see "Viewing the Organizational CA's properties and certificates" on page 75.

2 Start ConsoleOne.

3 Double-click the Organizational Certificate Authority object.

4 Click the Certificates tab.

5 Click the down-arrow to see the certificates available to view.

6 Click the Public Key Certificate.

This property page displays the fully-typed name of the subject, the issuer's fully-typed name, and the validity dates of the public key certificate.

7 To view additional information about an installed public key certificate, click Details.

The Details page displays information contained in the public key certificate on various tabs.

8 After you finish viewing the details, click Close, then click Cancel.

Viewing the CA's Self-Signed Certificate Properties

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

   To view the appropriate rights for this task, see "Viewing the Organizational CA's properties and certificates" on page 75.

2. Start ConsoleOne.
3. Double-click the Organizational Certificate Authority object.
4. Click Certificates.
5. Click the down-arrow to see the certificates available to view.
6. Click Self-Signed Certificate.

   The property page displays the subject's fully-typed name, the issuer's fully-typed name, and the validity dates of the self-signed certificate.

7. To view additional information about the certificate, click Details.

   The Details page displays information contained in the public key certificate on various tabs.

8. After you finish viewing the details, click Close, then click Cancel.

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 will accept a valid user or server certificate signed by the Organizational CA.

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

   To view the appropriate rights for this task, see "Exporting the Organizational CA's certificate(s)" on page 75.

2. Start ConsoleOne.
3. Double-click the Organizational Certificate Authority object.
4. Click the Certificates tab.
5. Click the down-arrow to see the available certificates.
6. Click Self-Signed Certificate.
7. Click Export.

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

8. When asked whether or not to export the private key, select No, then click Next.
9. Provide a filename and select a format that the certificate should be exported to (binary DER or text encoded base64).
10. Click Finish.

The certificate is saved to the file and is available to be imported into a cryptography-enabled application as the trusted root.

Backing Up an Organizational CA

If you have minted a significant number of certificates using your Organizational CA, you might want to 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 will be able to use the backup file to restore your Organizational CA to any server in the tree that has Certificate Server version 2.21 or higher installed.

NOTE: The ability to back up an Organizational CA is only available 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.

To back up and restore and Organizational CA:

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

   To view the appopriate rights for this task, see "Backing up and restoring an Organizational CA" on page 75.

2. Start ConsoleOne.
3. Double-click on the Organization Certificate Authority object.
4. Click the Certificates tab.
5. Click the down-arrow to see the available certificates.
6. Click either the Self-Signed Certificate or the Public Key Certificate. Both certificates will be written to the file during the backup operation.
7. Click Export.

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

8. When asked whether to export the private key, select Yes, then click Next.
9. Select the filename and the location for the backup file.
10. Specify a password with 6 or more alphanumeric characters to use in encrypting the PFX file.
11. Click Next.
12. Click Finish.

    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.

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 using a backup file created as described in “Backing Up an Organizational CA” on page 28.

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 NetWare 6 or later, the NICI configuration information is backed up by default using a backup utility.

To restore the Organizational CA:

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

   To view the appopriate rights for this task, see “Backing up and restoring an Organizational CA” on page 75.

2. Start ConsoleOne.
3. Delete the Organizational CA object if it exists.
4. Right-click the Security container object, then click New > Object.
5. From the list box in the New Object dialog box, double-click NDSPKI:Certificate Authority.

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

6. 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.
7. Specify the Import option.
8. Click Next.
9. Click Read from File, then select the name of the backup file in the dialog box.
10. Click Next.
11. Enter the password used to encrypt the file when the backup was made.
12. 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: If the backup file is no longer needed, the file and the media it was stored on should be destroyed.

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 28 and “Restoring an Organizational CA” on page 29.

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: If the backup file is no longer needed, the file and the media it was stored on should be destroyed.

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 using ConsoleOne. Any certificate in the NDS 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. The certificate chain for a certificate signed by your Organizational CA is composed of one certificate, which is the Organizational CA's self-signed certificate. Externally signed user and server certificates might have longer chains.

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

To view the appropriate rights for this task, see "Validating the Organizational CA's Certificates" on page 75.

2 Start ConsoleOne.

3 Double-click the Organizational CA.

4 Click Certificates, then click Self-Signed Certificate or Public Key Certificate.

5 Click Validate.

The Certificate Validation screen appears, providing the status of the certificate. If the certificate is not valid, the reason is given. Click Details for information about the exact certificate that was considered invalid.

6 Click OK to exit or click Details, if applicable, to view more information.

Replacing the Organizational CA

The private key and certificates in the Organizational CA object can be replaced. They can only be replaced using a PFX file created during a backup of an Organizational CA. The key and certificates in the file need not match the ones in the object; the data in the file will overwrite the key and certificates in the object.

Replacing the private key and certificates in the Organizational CA 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 Organizational CA and creating a new one. See the section “Deleting the Organizational CA” on page 31for more information on the consequences of deleting the Organizational CA.

If the key and certificates do match the ones in the object, replacing the Organizational CA will have no effect. If they match the key and certificates in a previous version of the Organizational CA, the only certificates adversely affected will be those signed by the CA whose key and certificates are being replaced. All of those certificates must be deleted and replaced with new ones signed by the new Organizational CA as described in the section “Deleting the Organizational CA” on page 31 for more information.

To replace the private key and certificates on the Organizational CA object:

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

   To view the appropriate rights for this task, see “Replacing the Organizational CA” on page 75.

2. Start ConsoleOne.
3. Double-click on the Organization Certificate Authority object.
4. Click the Certificates tab.
5. Click the down-arrow to see the available certificates.
6. Click either the Self-Signed Certificate or the Public Key Certificate.

   The operation can be started from either page. It will replace both certificates as well as the private key.

7. Click Replace.

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

8. Click Read from File, then select the name of the backup file in the dialog box.
9. Click Next.
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 replaced and the CA is fully functional. The backup file can be stored again for future use if desired.

IMPORTANT: If the backup file is no longer needed, the file and the media it was stored on should be destroyed.

Deleting the Organizational CA

Deleting the Organizational CA object should only be done if absolutely necessary or if you are restoring the Organizational CA from a backup (see “Restoring an Organizational CA” on page 29). 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 has become 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.

If the Organizational CA object must be deleted, use the following steps:

1. Using ConsoleOne, delete all Server Certificate objects that hold certificates signed by the Organizational CA.
2. Using ConsoleOne, delete all user certificates signed by the Organizational CA.

   NOTE: Some certificates may need to be archived to allow users to decrypt data at a later period.
3. The owners of any certificates issued to outside users or servers should be notified that the Organizational CA is being replaced.
4. Notify all external users, partners, and customers that the Organizational CA should no longer be trusted. The CA's certificate should be removed from lists of trusted authorities in applications, browsers, etc.
5. Using ConsoleOne, delete the Organizational CA object.
6. Create a new Organizational CA object.
7. Make a backup of the new Organizational CA.
8. Create new server certificates to replace each of those deleted in Step 1.
9. Create new user certificates, as needed, to replace those deleted in Step 2.
10. Issue new certificates for any outside users or servers to replace those invalidated in Step 3.
11. Publish the new Organizational CA's certificate and notify all external users, partners, and customers about the update.

Server Certificate Object Tasks

Creating Server Certificate Objects

This task is described in Chapter 2. See "Creating Server Certificate Objects" on page 19.

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 signed public key certificate to you.This task applies when you have created a Server Certificate object 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 will either return one or more files each containing one certificate, or 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 will contain 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 using Internet Explorer.

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 Novell Support.

To import the certificates into a Server Certificate object:

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

   To view the appropriate rights for this task, see "Importing a public key certificate into a Server Certificate object" on page 75.

2. Start ConsoleOne.
3. Double-click the Server Certificate object.
4. Click Import Certificates.

   The Import Certificates button is available from either of the pages on the Certificates tab.

5. Click Import Certificates.

   If all the certificates to be imported are in a single PKCS #7 file, check the No Trusted Root Available check box. Otherwise, do the following:

   - To paste the trusted root certificate into the dialog box, use any text editor to open the certificate you received from the Certificate Authority (CA), then copy the "-----BEGIN CERTIFICATE-----" and the "-----END CERTIFICATE-----" lines and all information between them. Then paste the certificate into the box provided in the Import Server Certificates dialog box.
   - To install the trusted root certificate from a file, click Read from File to browse for the certificate you received from the CA. Select the file, then click OK.

6. Click Next.
7. Indicate the location of the public key certificate you received from the Certificate Authority (CA).

   The public key certificate can be imported either by pasting it into the dialog box or by reading it from a file using the same procedure described in Step 5.

8. Click Finish.

   This stores all of the certificates entered into the wizard in the Server Certificate object. The Certificate property page now displays the distinguished names of the subject and issuer of the indicated certificate as well as the validity period of the certificate.

9. To view the details of your newly installed public key certificate, click Details. Click Help for further information about the certificate details page.
10. Click Close, then click Cancel.

    The Server Certificate object is now ready to be used by any cryptography-enabled application.

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.
- You will have 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 it can be pasted directly into a cryptography-enabled application.

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

   To view the appropriate rights for this task, see "Exporting a Trusted Root or Public Key Certificate from a Server Certificate object" on page 76.
2. Using ConsoleOne, double-click the Server Certificate object that the particular application is configured to use.
3. Click Certificates, and click the certificate you want to export.
4. Click Export.

   This opens a wizard that helps you export the certificate to a file.
5. When asked whether or not to export the private key, click No, then click Next.
6. Provide a filename and select a format that the certificate should be exported to (binary DER or text encoded base64).
7. Click Finish.
8. Use the file as needed.

   For example, if you want to install a trusted root certificate in an Internet Explorer 5.*x* 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 will automatically accept SSL connections with services that use certificates issued by this CA.

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 will not be able to 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.

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

   To view the appropriate rights for this task, see "Deleting a Server Certificate object" on page 76.
2. Start ConsoleOne.
3. Select the Server Certificate object that you want to delete.
4. Press the Delete key, then click Yes.

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.

These properties provide you with the information you need to perform any task related to this object.

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

   To view the appropriate rights for this task, see "Viewing the Server Certificate object's properties and certificates" on page 76.

2. Start ConsoleOne.
3. Double-click the Server Certificate object.

   This brings up the property pages for the Server Certificate Object, including a General page, a Certificates page, and property pages related to eDirectory.

4. Click each tab that you want to view.
5. Click Cancel.

Viewing a Server Certificate Object's Public Key Certificate Properties

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

   To view the appropriate rights for this task, see "Viewing the Server Certificate object's properties and certificates" on page 76.

2. Start ConsoleOne.
3. Double-click the Server Certificate object containing the public key certificate that you want to view.
4. Click the Certificates tab.
5. Click the down-arrow to see the certificates available to view.
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 additional information about a public key certificate, click Details.

   The Details page has various tabs displaying information contained in the public key certificate.

8. After you finish viewing the details, click Close > Cancel.

Viewing a Server Certificate Object's Trusted Root Certificate Properties

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

   To view the appropriate rights for this task, see "Viewing the Server Certificate object's properties and certificates" on page 76.

2. Start ConsoleOne.

3 Double-click the Server Certificate object containing the trusted root certificate that you want to view.

4 Click the Certificates tab.

5 Click the down-arrow to see the certificates available to view.

6 Click the 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, Novell Certificate Server indicates this.

7 To view additional information about an installed trusted root certificate, click Details.

The Details page has various tabs displaying information contained in the trusted root certificate.

8 After you finish viewing the details, click Close, then click Cancel.

Backing Up a Server Certificate Object

Novell 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 will be able to 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 backup 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. The process is as follows:

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

To view the appropriate rights for this task, see "Backing up and restoring a Server Certificate object" on page 76.

2 Start ConsoleOne.

3 Double-click the server certificate object.

4 Click the Certificates tab.

5 Click the down-arrow to see the available certificates.

6 Click either the Trusted Root Certificate or the Public Key Certificate. Both certificates will be written to the file during the backup operation.

7 Click Export.

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

8. When asked whether to export the private key, select Yes, then click Next.
9. Select the filename and the location for the backup file.
10. Specify a password with 6 or more alphanumeric characters to use in encrypting the PFX file.
11. Click Next.
12. Click Finish.

    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.

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 36.

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

NOTE: If you were unable to make a backup of the server certificate object, the server certificate object may still be usable if NICI 2.x is installed on the server and a backup was made of the NICI configuration information. See the NICI documentation for information on how to back up and restore the NICI configuration files.

To restore the server certificate object:

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

   To view the appropriate rights for this task, see "Backing up and restoring a Server Certificate object" on page 76.
2. Start ConsoleOne.
3. Delete the old server certificate object.
4. Right-click the container object and click New > Object.
5. From the list box in the New Object dialog box, double-click NDSPKI:Key Material.

   This opens the Create a Server Certificate Object dialog box and the corresponding wizard that creates the object.
6. In the creation dialog box, specify the server that should own the server certificate object and the key pair name of the server certificate. The server specified must have Certificate Server version 2.21 or higher installed and be up and running.
7. Specify the Import option.
8. Click Next.
9. Click Read from File, then select the name of the backup file in the dialog.
10. Click Next.
11. Enter the password used to encrypt the file when the backup was made.
12. 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 now be stored again for future use if desired.

IMPORTANT: If the backup file is no longer needed, the file and the media that it was stored on should be destroyed.

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 will 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. 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 "Creating Server Certificate Objects" on page 19.

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

   NOTE: If different services have different IP or DNS addresses, you will 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 identical key pair name as the first on all remaining servers in the cluster. See "Backing Up a Server Certificate Object" on page 36.

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 using ConsoleOne. 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. The certificate chain for a certificate signed by your Organizational CA is composed of one certificate, which is the Organizational CA's self-signed certificate. Externally signed user and server certificates might have longer chains.

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

To view the appropropriate rights for this task, see "Validating Server Certificates" on page 76.

2. Start ConsoleOne.
3. Double-click the Server certificate object that contains the certificate you want to validate.
4. Click Certificates, then click Trusted Root Certificate or Public Key Certificate.
5. Click Validate.

   The Certificate Validation screen appears, providing the status of the certificate. If the certificate is not valid, the reason is given. Click Details for information about the exact certificate that was considered invalid.

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 36 and "Restoring a Server Certificate Object" on page 37.

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: If the backup file is no longer needed, the file and the media that it was stored on should be destroyed.

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 a 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 will overwrite 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 34 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 will have 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. Log in to the eDirectory tree as an administrator with the appropriate rights.

   To view the appropropriate rights for this task, see "Replacing a server certificate's keying material" on page 76.

2. Start ConsoleOne.
3. Double-click the server certificate object.
4. Click the Certificates tab.
5. Click the down-arrow to see the available certificates.

6 Click either the Trusted Root Certificate or the Public Key Certificate.

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

7 Click Replace.

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

8 Click Read from File, then select the name of the backup file in the dialog.

9 Click Next.

10 Enter the password used to encrypt the file when the backup was made.

11 Click Finish.

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

IMPORTANT: If the backup file is no longer needed, the file and the media that it was stored on should be destroyed.

User Certificate Tasks

Creating User Certificates

This task is described in Chapter 2. See "Create a User Certificate" on page 20.

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.

NOTE: In order for the user certificates to be created, each User object must have an e-mail address listed in the User object properties.

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

To view the appropriate rights for this task, see "Creating user certificates" on page 76.

2 In ConsoleOne, select multiple User objects or select the container that the User objects reside in.

3 Click File > Properties of Multiple Objects.

If you selected a container that contains eDirectory objects other than User objects, you are prompted to select an Object class. Select the User object class, then click OK.

You can change the list by using the Add or Remove buttons on this page. You can add users from other containers by using the Add button.

4 Click the Security tab > Certificates.

5 Click Create.

This launches the certificate creation wizard, which will guide you through the creation process. For specific information on the wizard pages, click Help.

Importing a Public Key Certificate into a User Object

You can import any public key certificate into a user object (for example, a certificate signed by a third-party Certificate Authority). Once imported, the certificate is stored in the User object and appears on the list of certificates available.

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

   To view the appropriate right for this task, see "Importing a public key certificate into a User object" on page 76.

2. Start ConsoleOne.
3. Double-click the User object that you want to host the imported certificate.
4. Click the Security tab > Certificates.
5. Click Import.
6. Enter 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 certificate to import.

   - To paste the certificate into the dialog box, use any text editor to open the certificate, then copy the "-----BEGIN CERTIFICATE-----" and the "-----END CERTIFICATE-----" lines and all information between them. Then paste the certificate in the box provided in the Import User Certificates dialog box.
   - To install the certificate from a file, click Read from File to browse for the certificate. Select the file, then click OK.

8. Click Finish.

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

NOTE: Private keys cannot be imported to the User object.

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.

These properties provide you with the information you need to perform any task related to this object.

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

   To view the appropriate rights for this task, see "Viewing a user certificate's properties" on page 76.

2. Start ConsoleOne.
3. Double-click the User object that hosts the user certificate.

   This brings up the property pages for the User object.

4. Click the Security tab > Certificates.

5 Click a certificate to view its properties.

6 Click Close, then click Cancel.

Exporting a User Certificate Using ConsoleOne

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 ConsoleOne. The other person's certificate can also be obtained using LDAP or e-mail.

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

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

To view the appropriate rights for this task, see "Exporting a user certificate using ConsoleOne™" on page 76.

2 Start ConsoleOne.

3 Double-click the User object that hosts the user certificate.

4 Click Security > Certificates tab.

5 Click the user certificate that you want to export.

6 Click Export.

This opens a wizard that helps you export the user certificate to a file. If you are not logged in as the user that owns the certificate, you will not be able to export the private key. If you are logged in as that user, you will be asked whether to export the private key as well -- select No. See "Exporting a User Certificate and Private Key Using ConsoleOne" on page 42.

Exporting a User Certificate and Private Key Using ConsoleOne

In order to use a certificate for secure e-mail, authentication, or encryption, you must export both the private key and the certificate. Knowing the private key proves that you are the person indicated in the certificate.

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

To export your own private key and certificate:

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

To view the appropriate rights for this task, see "Exporting a user's private key and certificate using ConsoleOne" on page 76.

2 Start ConsoleOne.

3 Double-click the User object that hosts the user certificate.

4 Click the Security tab > Certificates.

5 Click the user certificate that you want to export.

6 Click Export.

This opens a wizard that helps you export the user certificate to a file. If you are not logged in as the user who owns the certificate, you will not be able to export the private key. If you are logged in as that user, you will be asked whether to export the private key as well. Select Yes.

7 Select the filename and the location for the backup file.

8 Specify a password with 6 or more alphanumeric characters to use in encrypting the PFX file.

9 Click Next.

10 Click Finish.

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.

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.

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

To view the appropriate rights for this task, see "Deleting a user certificate and private key" on page 76.

2 Start ConsoleOne.

3 Double-click the User object that hosts the user certificate. This brings up the property pages for the User object.

4 Click the Security tab > Certificates.

5 Click the certificate you want to delete.

6 Click Delete.

7 Click Yes to verify that you want to delete the user certificate and private key.

8 Click Cancel.

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 using ConsoleOne. 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. The certificate chain for a certificate signed by your Organizational CA is composed of one certificate, which is the Organizational CA's self-signed certificate. Externally signed user and server certificates might have longer chains.

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

   To view the appropriate rights for this task, see "Validating User Certificates" on page 76.
2. Start ConsoleOne.
3. Double-click the User object that hosts the certificate you want to validate.
4. Click Security > Certificates tab.
5. Select the User certificate you want to validate.
6. Click Validate.

   The Certificate Validation screen appears, providing the status of the certificate.
7. Click OK to exit.

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 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 44 and "Creating a Trusted Root Object" on page 44.

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.

Trusted Root Object Tasks

Creating a Trusted Root Container

This task is described in Chapter 2. See "Creating a Trusted Root Container" on page 21.

Creating a Trusted Root Object

This task is described in Chapter 2. See "Creating Trusted Root Objects" on page 21.

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.

These properties provide you with the information you need to perform any task related to this object.

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

   To view the appropriate rights for this task, see "Viewing a Trusted Root object's properties" on page 76.
2. Start ConsoleOne.

3 Open the Trusted Root Container that hosts the Trusted Root object.

4 Double-click the Trusted Root object.

This brings up the property pages for the Trusted Root object.

5 Click each tab that you want to view.

6 Click Cancel.

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

To view the appropriate rights for this task, see "Replacing a trusted root certificate" on page 77.

2 Start ConsoleOne.

3 Open the Trusted Root Container that hosts the Trusted Root object.

4 Double-click the Trusted Root object.

This brings up the property pages for the Trusted Root object.

5 Click the Trusted Root tab.

6 Click Replace.

This opens the Replace a Trusted Root Certificate Wizard that helps you replace the Trusted Root Certificate. For specific information on the wizard pages, click Help.

7 Click Cancel.

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 using ConsoleOne. 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. The certificate chain for a certificate signed by your Organizational CA is composed of one certificate, which is the Organizational CA's self-signed certificate. Externally signed user and server certificates may have longer chains.

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

   To view the appropriate rights for this task, see "Validating a trusted root certificate" on page 77.

2. Start ConsoleOne.
3. Double-click the trusted root object that hosts the certificate you want to validate.
4. Click the Trusted Root tab.
5. Click Validate.

   The Certificate Validation screen will appear, providing the status of the certificate.

6. Click OK to exit.

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 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 44 and "Creating a Trusted Root Object" on page 44.

Certificate Revocation List (CRL) Tasks

Creating a Certificate Revocation List (CRL) Object

This task allows you to create a CRL Distribution Point object in eDirectory. This object can be created in any container in the eDirectory tree. As part of the creation process, you will be asked to provide a CRL. You will need to obtain a CRL from a third-party CA. If you don't have a CRL file at the time you create the CRL Distribution Point object, you can still create the object and import the CRL later.

To create a CRL Distribution Point object using ConsoleOne:

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

   To view the appropriate rights for this task, see "Creating a CRL Object" on page 77.

2. Start ConsoleOne.
3. Right-click in any container in the tree > click New > Object > cRLDistributionPoint, then click OK.
4. Type a name for the object, then click Next.
5. Paste a copy of the CRL into the field or read it from a CRL file.
6. Click Finish to create the object.

To create a CRL Distribution Point object using Novell iManager:

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

   To view the appropriate rights for this task, see "Creating a CRL Object" on page 77.

3. From the Roles and Tasks menu, click PKI Certificate Management > Create CRL Distribution Point.

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 OK to create the object.

Importing a Third-Party CRL

This task allows you to import a CRL signed by a third-party certificate authority into a CRL Distribution Point object. This option is only active if no CRL is present in the object.

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

To view the appropriate rights for this task, see "Importing a third-party CRL" on page 77.

2 Start ConsoleOne.

3 Double-click the CRL Distribution Point object that you want to import the CRL into.

4 Click Import.

If the Import button is not active, it means that this CRL Distribution Point object already contains a CRL. You can replace the existing CRL by clicking Replace.

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

6 Click Finish.

Exporting a Third-Party CRL

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

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

To view the appropriate rights for this task, see "Exporting a third-party CRL" on page 77.

2 Start ConsoleOne.

3 Double-click the CRL Distribution Point object that you want to export the CRL from.

4 Click Export.

If the Export button is not active, it means that this CRL Distribution Point object does not contain a CRL. You can import a CRL by clicking Import.

5 Select the format you want to save the CRL to (binary encoded DER or text encoded Base64), then specify a filename.

The extension for the file is .crl by default. You can also browse to select the location that the file will be saved to.

6 Click Export.

Replacing a Third-Party CRL

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

To view the appropriate rights of this task, see "Replacing a third-party CRL" on page 77.

2 Start ConsoleOne.

3 Double-click the CRL Distribution Point object that contains the CRL you want to replace.

4 Click Replace.

If the Replace button is not active, it means that this CRL Distribution Point object does not contain a CRL. You can import a CRL by clicking Import.

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

6 Click Finish.

Viewing a Third-Party CRL's Properties

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

To view the appropriate rights for this task, see "Viewing a third-party CRL" on page 77.

2 Start ConsoleOne.

3 Double-click CRL Distribution Point object.

eDirectory Tasks

Merging Two Trees that Have Security Containers

Special considerations need to be made when merging eDirectory trees when a Security container has been installed in one or both of the trees. Make sure that this is something you really want to do; this procedure has the potential to be a very time-consuming and laborious task.

IMPORTANT: These instructions are complete for trees with Novell Certificate Server version 2.21 and earlier, Novell Single Sign-on version 1.*x*, and NMAS version 1.*x*.

1 In ConsoleOne, identify the trees to be merged.

2 Identify which tree will be the source tree and which tree will be the target tree.

The security considerations to keep in mind when choosing which tree will be the source and which tree will be the target are as follows:

- Any certificates signed by the source tree's Organizational CA must be deleted.
- The source tree's Organizational CA must be deleted.
- All user secrets stored in Secret Store on the source tree must be deleted.
- All NMAS login methods in the source tree must be deleted and reinstalled in the target tree.
- All NMAS users that were in the source tree must be re-enrolled when the trees are merged.
- All users and servers that were in the source tree must have new certificates created for them when the trees are merged.
- All users that were in the source tree must have their secrets reinstalled into their Secret Stores.

3 If neither the source tree nor the target tree has a container named Security under the [Root] of the tree, or if only one of the trees has the Security container, no further action is required. Otherwise, continue with the remaining procedures in this section.

Novell Certificate Server

If Novell Certificate Server has been installed on any server in the source tree, the following steps should be performed. Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, skip the step.

NOTE: Previous version of Novell Certificate Server were called Public Key Infrastructure Services (PKIS).

1. Any Trusted Root certificates in the source tree should be installed in the target tree. Trusted Root certificates are stored in Trusted Root objects which are contained by Trusted Root containers. Trusted Root containers can be created anywhere within the tree; however, only the Trusted Root certificates that are in the Trusted Root containers within the Security container must be moved manually from the source tree to the target tree.
2. Install the Trusted Root certificates in the target tree.
   - **2a** Pick a Trusted Root container in the Security container in the source tree.
   - **2b** Create a Trusted Root container in the Security container of the target tree with the exact name used in the source tree (Step 2a).
   - **2c** In the source tree, open a Trusted Root object in the selected Trusted Root container and export the certificate. Remember the location and filename you choose, so you can use it in the next step.
   - **2d** In the target tree, create a Trusted Root object in the container created in Step 2b. Specify the same name as the source tree and, when prompted for the certificate, specify the file you created in Step 2c.
   - **2e** Delete the Trusted Root object in the source tree.
   - **2f** Repeat Step 2c through Step 2e until all Trusted Root objects in the selected Trust Root container have been installed into the target tree.
   - **2g** Delete the Trusted Root container in the source tree.
   - **2h** Repeat steps Step 2a through Step 2f until all Trusted Root containers have been deleted in the source tree.
3. Delete the Organizational CA in the source tree. The Organizational CA object is in the Security container.

   NOTE: Any certificates signed by the Organizational CA of the source tree will be invalid following Step 3. This includes server certificates and user certificates that have been signed by the Organizational CA of the source tree. The certificates might continue to work, but deleting the Organizational CA that issued them effectively invalidated them. Therefore, the certificates must be reissued.
4. Delete every Key Material object (KMO) in the source tree which has a certificate signed by the Organizational CA of the source tree.

   Key Material objects in the source tree with certificates signed by other CAs will continue to be valid and need not be deleted.

   NOTE: If you are uncertain about the identify of the signing CA for any Key Material object, check the Trusted Root Certificate section of the Certificates tab in the Key Material object properties page.
5. Delete all user certificates in the source tree which have been signed by the Organizational CA of the source tree.

   If users in the source tree have already exported their certificates and private keys, those exported certificates and keys will continue to be usable, but they should be replaced by new

certificates signed by the target tree's Organizational CA. Private keys and certificates that are still in eDirectory will cease to be exportable after you perform Step 3 above.

For each user with certificates, open the Properties of the user object. Under the Certificates section of the Security tab, a table will list all the certificates for the user. All of those certificates with the Organizational CA as the issuer must be deleted.

User certificates will only be present in the source tree if Novell Certificate Server version 2.0 or later has been installed on the server which hosts the Organizational CA in the source tree.

Novell Single Sign-on

If Novell Single Sign-on has been installed on any server in the source tree, the following step should be performed. Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, skip the step.

1. Delete all Novell Single Sign-on secrets for users in the source tree.

   For every user using Novell Single Sign-on in the source tree, open the Properties of the user object. All of the user's secrets will be listed under the Secret Store section of the of the Security tab. Delete all listed secrets.

NMAS

If NMAS has been installed on any server in the source tree, the following steps should be performed. Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, skip these steps.

1. In the target tree, install any NMAS login methods that were in the source tree but not in the target tree.

   NOTE: To ensure that all of the necessary client and server login components are properly installed in the target tree, we recommend that all new login methods be reinstalled using original Novell or vendor-supplied sources. (Although methods can be reinstalled from existing server files, establishing a clean install from Novell or vendor-supplied packages is usually simpler and more reliable.)

2. To ensure that the previously established login sequences in the source tree are available in the target tree, migrate the desired login sequences.

   2a In ConsoleOne, select the Security container in the source tree.

   2b Right-click the Login Policy object and select Properties.

   2c For each login sequence listed in the Defined Login Sequences drop-down menu, note the login methods used (listed in the right pane).

   2d Select the Security container in the target tree and replicate the login sequences using the same login methods noted in Step 2c.

   2e Click OK when you are finished.

3. Delete NMAS login security attributes in the source tree.

   3a In the Security container of the source tree, delete the Login Policy object.

   3b In the Authorized Login Methods container of the source tree, delete all login methods.

   3c Delete the Authorized Login Methods container in the source tree.

Novell Security Domain Infrastructure

If Novell Certificate Server, Novell Single Sign-on, or NMAS has been installed on any server in the source tree, the Novell Security Domain Infrastructure (SDI) will be installed. If SDI has been installed, the following steps should be performed. Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, skip the step.

1. Delete the W0 object and then the KAP container in the source tree. The KAP container is in the Security container. The W0 object is in the KAP container.
2. On all servers in the source tree, delete the Security Domain Infrastructure (SDI) keys by deleting the sys:\system\nici\nicisdi.key file.

   IMPORTANT: This deletion should be performed on all servers in the source tree.

Other Security-Specific Operations

If a Security container exists in the source tree, delete it before you merge the trees.

NOTE: Depending on how the product was used, the objects and items referred to might or might not be present. If the objects and items referred to are not present in the source tree, skip the step.

1. For any remaining objects in the Security container of the source tree, consult the product documentation for instructions on how to safely move the information contained in the object to the target tree.
2. Delete the remaining objects in the Security container of the source tree.
3. Delete the Security container in the source tree.

Performing the Tree Merge

eDirectory trees are merged with the DSMERGE utility. For more information, refer to the DSMERGE documentation (http://www.novell.com/documentation/lg/nw51/docui/index.html#../utlrfenu/data/hv7lr8sz.html).

Product-Specific Operations to Perform After the Tree Merge

Novell Security Domain Infrastructure

If the W0 object existed in the target tree before the merge, the Security Domain Infrastructure (SDI) keys used by the servers that formerly resided in the target tree must be installed in the servers that formerly resided in the source tree.

The easiest way to accomplish this is to install Novell Certificate Server 2.0 or later on all servers formerly in the source tree that held SDI keys (sys:\system\nici\nicisdi.key file). This should be done even if the Novell Certificate Server has already been installed on the server.

If the W0 object did not exist in the target tree before the merge but did exist in the source tree, the SDI must be reinstalled on the resulting tree.

The easiest way to accomplish this is to install Novell Certificate Server 2.0 or greater on the servers in the resulting tree. Novell Certificate Server must be installed on the servers formerly in the source tree that held SDI keys (sys:\system\nici\nicisdi.key file). It can also be installed on other servers in the resulting tree.

Novell Certificate Server

1. Reissue certificates for servers and users formerly in the source tree, as necessary.

NOTE: We recommend that all servers that hold a replica of the partition containing a user's object have Novell Certificate Server version 2.0 or later installed.

In order to issue a certificate for a user, Novell Certificate Server version 2.0 or later must be installed.

The server that hosts the Organizational CA must have Novell Certificate Server version 2.0 or later installed.

Novell Single Sign-on

1. Re-create Secret Store secrets for users who were formerly in the source tree, as necessary.

NMAS

1. Reinstall the NMAS methods that were formerly in the source tree, as necessary.
2. Re-enroll NMAS users who were formerly in the source tree, as necessary.

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

Novell Certificate Server can be installed on multiple servers in an eDirectory tree. However, for Novell 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 Novell Certificate Server on multiple servers in an eDirectory tree, you must allow eDirectory to replicate between each installation of Novell 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 Novell 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 Novell 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*.

Restoring or Re-creating a Security Container

If you delete the Security container, you will not be able to 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:

- Log in as a network administrator with Create rights at the root of the eDirectory tree. Start ConsoleOne. Right-click on the Root container and click New > Object. From the list box in the New Object dialog box, double-click SAS:Security. The container name must be Security.
- Reinstall Novell Certificate Server on any server in the eDirectory tree.

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://www.support.novell.com/search/kb_index.htm) and search for TID #10053572, How to Restore or Recreate KAP and W0 Objects, for information on how to resolve this problem. You should not attempt further installations of Novell Certificate Server, Single Sign-on, NMAS, NetWare or eDirectory until the problems have been corrected.

Application Tasks

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

For cryptography-enabled applications not mentioned in this section, we recommend that you consult the application's documentation for specific instructions.

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

1. Export your Organizational CA's self-signed certificate, your user certificate, and the matching private key to a .pfx file.
2. Import the .pfx file into your e-mail client.
3. Configure your e-mail client to secure your e-mail.

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 certficate 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 or enter 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 Outlook98.

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 the filename of the exported .pfx file.
7. Type the password you used to protect the .pfx file.
8. Click OK.

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

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 would like 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, click the check box next to Sign Digitally. To enable encryption as the default for all outgoing email, click 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 will be 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 would like 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, click 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, check the check box next to Encrypt for Recipient Using. You can 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. Check 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.

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 will 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.

NOTE: 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 27. Then import it into your browser according to the directions below.

NOTE: The following Internet browsers will only recognize certificates that have been exported in .DER or a .CRT format. Although .B64 is an optional export format, it will not be recognized by these Internet browsers.

Microsoft Internet Explorer Version 4

If you are using Microsoft Internet Explorer version 4, complete the following to import the Organizational CA's certificate:

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

   This opens the New Site Certificate dialog.
4. Under Available Usages, check the check box next to Secure E-Mail, then click OK.
5. Click Yes to add the certificate to the Root Store.

Microsoft Internet Explorer Version 5

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. Enter or browse for the filename of the exported Organizational CA's self-signed certificate, then click OK.

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

   This opens the Certificate Manager Import Wizard.
5. Click Next.

6 Select the area where you would like to store the certificate, then 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 a NetWare server, this file is located at sys:\public\mgmt. 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 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 checked.

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 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 checked.

6 Click Next until the dialog box to enter a short name for this Certificate Authority appears.

7 Click Finish.

Configuring Microsoft Internet Explorer (IE) for SSL with Novell Certificates

To configure IE to use Novell 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 57. Otherwise, any attempt to use IE to connect to a server that is using Novell certificates for SSL will only display an error.

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

Configuring Microsoft IIS for Client Authentication with Novell Certificates

To perform client authentication to IIS with Novell 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.
3. Check 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. Before clicking Finish, verify that the summary displays Certificate Store Selected by User and Trusted Root Certification Authorities/Local Computer.
6. Stop and restart the IIS services after installing your Organizational CA certificate.

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

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. 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 25.

However, you must first 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 Novell Certificate Server. Refer to the IIS online documentation for further instructions on installing the resulting server certificate and configuring IIS for SSL.

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://www.support.novell.com).

Installation Issues

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.

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 Novell Certificate Server™ on a server whose name is not listed by typing the name of the server in the text box.

Error Creating SAS Service Object During Install

When installing Novell 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 Novell Single Sign-on, NMAS, or Novell 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 will fail and a message will be 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 Novell Single Sign-on, NMAS, and Novell 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 ConsoleOne®. 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.

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.

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 ConsoleOne to create an Organizational CA and server certificates and export the trusted root.

Installation Fails with a -1443 Error

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

User Certificate Issues

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.

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.

-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 may 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.

Workstation Cryptography Strength

If U.S/Worldwide (high-grade) cryptography is loaded on your NetWare® server, you will have the option to create user certificates with key sizes of 512, 768, 1024, and 2048 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.

Server Certificate Issues

External CAs

Some third-party CAs like 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.

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 must also be moved.

DNS Support

If NetWare 5 Support Pack 3 or later is installed and 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 will be the fully distinguished name of the server. You can modify the default subject name by selecting Custom during the certificate creation process.

NOTE: DNS was not available prior to NetWare 5 Support Pack 3.

Deletion of the SAS Object

If you delete the SAS service object, any server certificates previously created for that server cannot be used by applications on the server. If these certificates are still needed, you can restore the SAS object from backup. If that is not possible, contact Novell Support for assistance.

Removing a Server from a Tree

There are several Certificate Server issues you need to consider when you remove a server from an eDirectory tree. Go to the Novell Support Web site (http://www.support.novell.com/search/kb_index.htm) and search for TID #10056795, Certificate Server Issues - Removing a Server.

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

Some external Certificate Authorities provide certificates that enable 40- 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- or 56-bit cryptography will experience unrecoverable SSL errors when communicating with servers configured with these certificates.

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

Subject Name Limitations for CAs

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

ConsoleOne Issues

ConsoleOne on a Server

You cannot manage Novell Certificate Server with ConsoleOne running on a server. You must run ConsoleOne from a Windows 95/98, Windows 2000, or Windows NT client workstation.

Refresh After Update

Occasionally, after you delete a user certificate, the ConsoleOne display does not refresh. To force the refresh, close the object view and reopen it.

Organizational CA Issues

Path Length

You can enter a maximum value of 255 for the path length of the Organizational CA. Any value greater than that will result in an error.

Moving the Organizational CA Object

The Organizational CA object must reside in the Security container.

Validation Issues

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. The certificate chain for a certificate signed by your Organizational CA is composed of one certificate, which is the Organizational CA's self-signed certificate. Externally-signed user and server certificates might have longer chains.

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:

1. 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.

2. 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 doing one or more of the following:

- Partition the Security container and widely replicate Read-Only replicas of it. The data in the Security container is relatively static, so doing so will not lead to synchronization problems within your tree.
- Place the replicas of the Security container physically closer to those users who need to validate user certificates.
- Remove expired certificates that are no longer trusted from the Trusted Roots container.

Validating Certificates after Deleting the Organizational CA

If you delete the Organizational CA (other than during a backup and restore procedure), you should delete all user and server certificates that were signed by the Organizational CA. If you don't, you will experience the following behavior when validating these certificates:

- User certificates signed by the deleted CA will be invalid. This is because the certificate of the CA which 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 will 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 delete all user and server certificates that were signed by the CA. You should also tell all users who may have imported your Organizational CA's certificate into their browsers to delete the certificate.

Miscellaneous Issues

-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 Novell Support.

Renaming the Security Container

You cannot rename the security container.

Certificate Encodings

UTF8 encoding for subject and issuer names in certificates is not supported. Certificate Server cannot issue or use certificates with this type of encoding.

Although Certificate Server can issue certificates with subject names and issuer names encoded in Unicode, you might encounter problems using them. You might be unable to establish an SSL connection using server certificates with Unicode subject or issuer names.

A Public Key Cryptography Basics

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.

Secure Transmissions

Data transmissions are private and secure when two things happen:

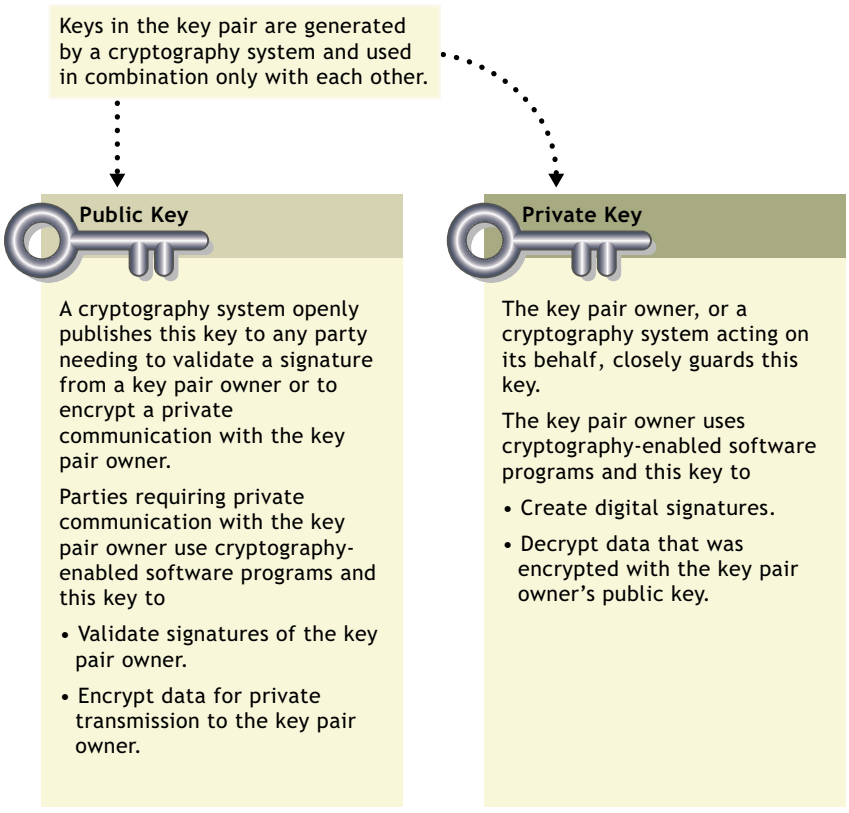
- **Authentication**—The data receiver knows that the data sender is exactly who or what it claims to be.
- **Encryption**—The data sent is encrypted so that it can be read only by the intended receiver.

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 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.

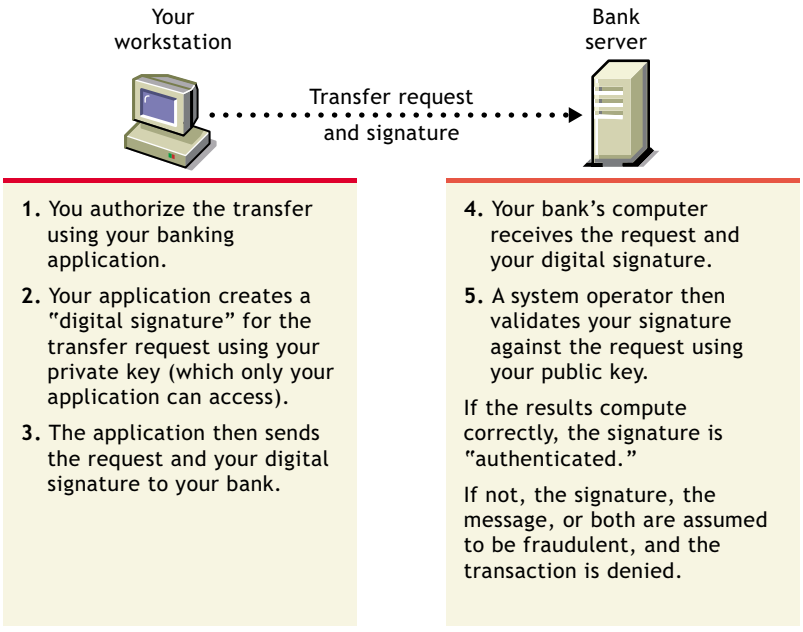
The basic principles and functions of each key in the key pair are summarized in the following illustration.



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.



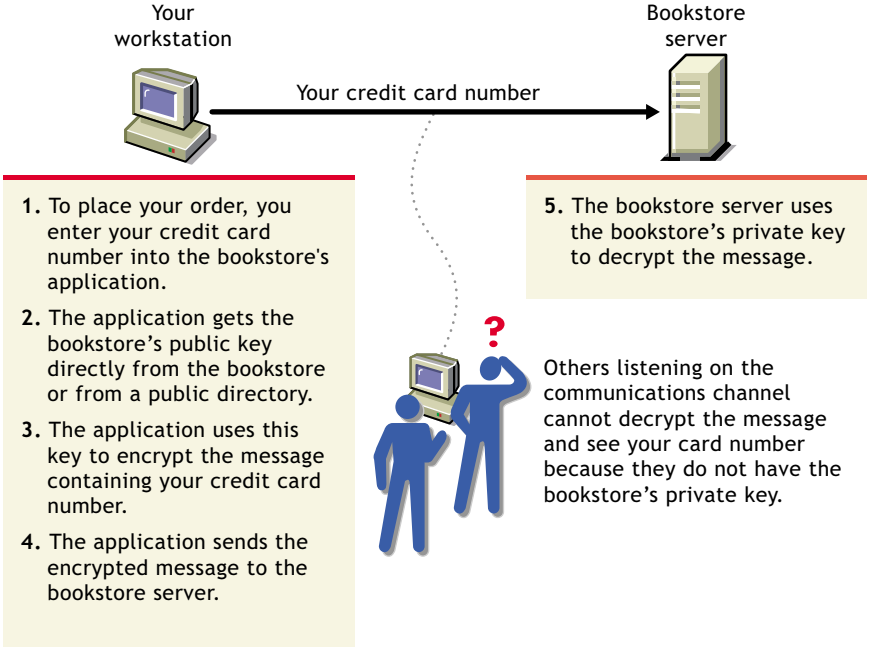
For information about digital signatures and their verification, see "Digital Signatures" on page 73.

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.



Establishing Trust

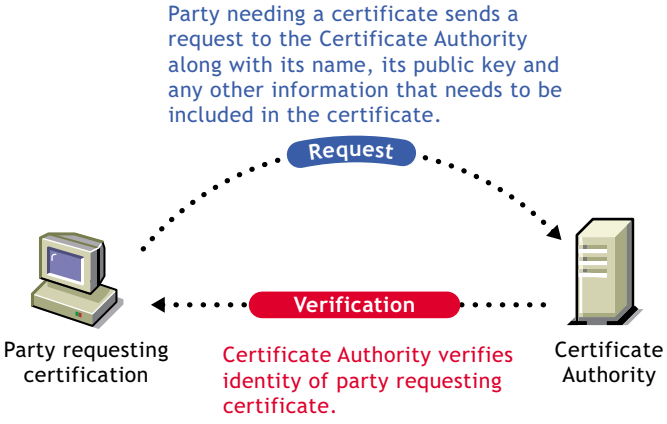
If a sender and receiver know and trust each other, they can simply exchange public keys and establish secure data transmission, including authentication and encryption. To do this, they would use each other's public keys and their own private keys.

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.

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.

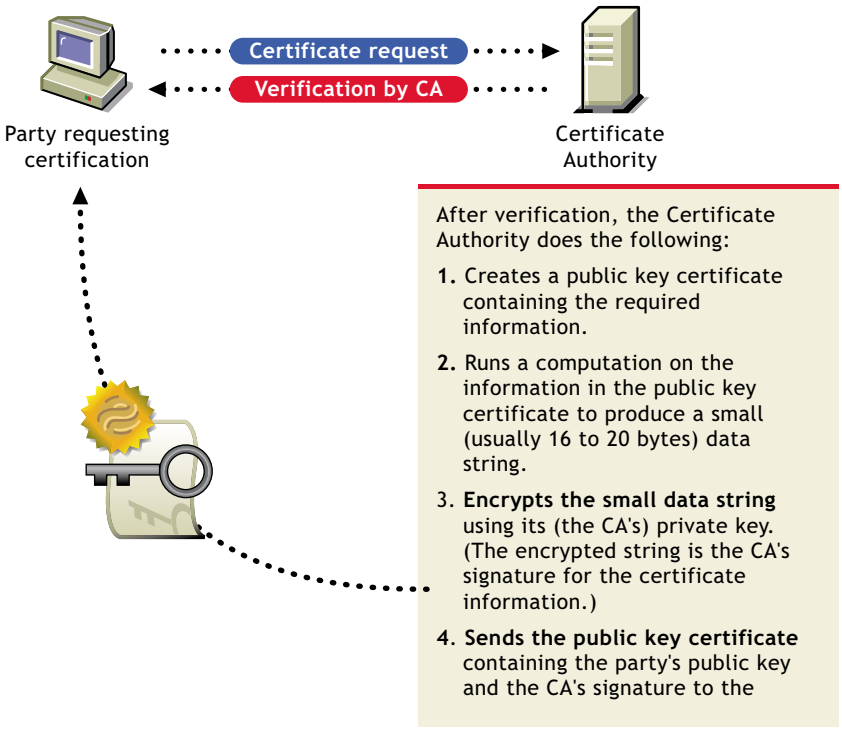


Once the identity of the requesting party has been established to the satisfaction of the certificate authority, the certificate authority's issues an electronic "certificate" and applies its digital signature.

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.



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 will produce identical results. It can then be safely assumed that neither the data nor the signature was modified in transit.

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 (i.e., 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 will be longer. The first two elements would still be the end entity certificate, followed by the certificate of the intermediate CA. But, the intermediate CA's certificate would then be followed by the certificate of its CA. This listing would then continue until the last certificate in the list was for a root CA. Thus, a certificate chain can be infinitely long. In practicality, most certificate chains have only two or three certificates, though.

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

This listing provides the specific entry rights an administrator needs to manage Novell® Certificate Server tasks within an eDirectory® tree. These rights are the minimum entry rights needed.

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

Tasks	Entry Rights Needed
Install Novell Certificate Server	For the first installation to an NDS tree: ◆ Supervisor at the [Root] of the tree For subsequent installations: ◆ Supervisor to the W0 object
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
Backing up and restoring an Organizational CA	◆ Supervisor on the Organizational CA's object
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 Org. CA)
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

Tasks	Entry Rights Needed
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
Replacing a server certificate's keying material	◆ Write to the attribute *NDSPKI:PrivateKey* on the server certificate object
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
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 using ConsoleOne™	◆ Browse on the User object
Exporting a user's private key and certificate using ConsoleOne	◆ 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
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
Deleting a Trusted Root object	◆ Delete on the Trusted Root object
Creating a CRL Object	◆ Create to the container that the *cRLDistributionPoint* object will be created in
Importing a third-party CRL	◆ Write to the attribute *certificateRevocationList*
Exporting a third-party CRL	◆ Read from the attribute *certificateRevocationList*
Replacing a third-party CRL	◆ Browse on the CRL object
Viewing a third-party CRL	◆ Browse to the attribute *certificateRevocationList*
Creating a Security container	◆ Create at the root of the eDirectory tree
Creating a SAS service object	◆ Supervisor on the object's container ◆ Write to the attribue *SAS:Service DN* on the server that the object is being created